Treasury Architecture Development Process

Version 1.0

Office of the
Deputy Assistant Secretary
for Information Systems
and
Chief Information Officer

30 September 1997

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont)

TABLE OF CONTENTS (Cont)

TABLE OF CONTENTS (Cont)

FIGURES

TABLE OF CONTENTS (Cont)

FIGURES (Cont)

TABLES

SECTION 1
INTRODUCTION

This document presents the architecture development process for developing an enterprise information system architecture (EISA) according to the Treasury Information Systems Architecture Framework (TISAF).

1.1 PURPOSE

The Treasury Architecture Development Process (TADP) describes a process for the specification, design, development, and implementation of specific information system architecture solutions that are consistent with the TISAF.

This document is part of the Department's commitment to establish guiding principles, promote standards usage, and provide technical guidance, which is fulfilled with the publication of this document.

1.2 BACKGROUND

Most software today is still produced using a custom-crafted approach. Many frameworks, toolkits, and collections of reusable components are becoming available for constructing software applications, but are not widely used. Consequently, systems continue to be expensive to design and build, often take longer to develop than they should, and often do not fully satisfy the business user's needs. Recent developments in the areas of process methodologies, specification languages, application generators, development environments, and programming tools have focused on improving the design and development of software.

As information systems become more complex, the overall system structure—or system architecture—becomes a central design problem. Design issues at this level include the high-level organization of the information system, its control and data structures, the assignment of functionality to logical and physical computational units, and the high-level interactions between these computational units.

During fiscal year (FY) 1997, the Office of the Deputy Assistant Secretary for Information Systems (DASIS) and Chief Information Officer (CIO) developed the TISAF to provide guidance concerning the development and evolution of information system architectures to meet the needs of business operations. The TISAF is a component of the Department's information resources management (IRM) strategy.

To assist the Treasury bureaus in applying the TISAF methodology to the development of information system architectures, an architectural development process is provided in this document to help the bureaus design and implement an EISA.

1.2.1 SYSTEM ARCHITECTING

System architecting is a multidisciplinary approach to defining and developing an architecture for an organization (Rechtin 1997). The system architect is responsible for establishing a satisfactory and feasible system concept at the earliest stage of system development, for maintaining the integrity of that concept throughout development, and for certifying the fitness of the resulting system for use by the client or customer. System architecting is a process for describing information system architectures in sufficient detail to make them useful to develop information systems.

An EISA is a description of a set of information system architectures. If the structure of the EISA is regular and clear, then it can be reused to describe new information systems or drive the modernization of existing systems.

Systems architecting is driven by the need to satisfy the client's requirements. A successful information system satisfies a useful purpose at an affordable cost for an acceptable period of time. Several benefits accrue from applying an architectural approach to EISA development.

Mission Focus

System architecting looks beyond the needs of a single information system. While traditional analysis focuses on solving a particular problem, system architecting looks across the business operations. By creating models describing an organization's business operations and the environment in which they operate, system architecting permits the specification of families of systems. By focusing the development methodology on the business operations, information systems functionality provided by the EISA can be aligned with the organization's missions.

Requirements Focus

A constant challenge for information systems development groups is coming to agreement on system requirements—both business and functional. Gathering requirements is often difficult because discussions are vague and unbounded. Using the system architecture as a starting point, developers and customers limit their conversations to agreed upon levels of detail and areas of concern. Once consensus is reached on one level of requirements in one area, the analysis may either drill down to the next level of detail or move laterally to another set of concerns.

Development Focus

A continuing challenge for information systems organizations is ensuring delivery of information systems that satisfy the business and functional requirements. An organization wants to be certain that the system, as developed, will conform to its requirements, and it wants to be able to reliably estimate and manage the work to be done. A system architecture guides both the requirements gathering and system development processes. Development projects progressively elaborate the architecture, adding more detail over time. When requirements and development share a common terminology and set of descriptions, there is an assurance that the system, as constructed, will conform to the requirements. And because development is more elaboration than invention, planning and managing the work can be more reliable. Although problems may still arise with

respect to both requirements and implementation, a system architecture guides problem resolution. When things go wrong, developers can use the system architecture to locate the areas of concern and level of detail and proceed from there.

Ability to Adapt

In the same way that an architecture aids in constructing a system, it helps ensure its evolution. The horizontal and vertical partitions permit localized changes without wide-ranging ramifications. When it is time to discuss changes, the system architecture guides the discussion, and assists in selecting the proper concerns and level of detail. The architecture allows information systems to gracefully evolve over time, handling requirements that were unknown when the system architecture was initially drafted. Being grounded in the business domain rather than application requirements is the key to a system architecture's longevity.

1.2.2 NEED FOR FLEXIBILITY

External and internal events will continually drive changes in departmental architectures. As a result, the architectures must evolve in several ways:

- New systems, technologies, and techniques for sharing and using information based on business case development and process and technology requirements analyses will be implemented.

- Reengineered business processes and systems that have been revamped using structured methodologies to increase their value-added merit and leverage new technologies and prior investments will require enhancements to existing systems.

- Adapted legacy systems will be rendered more open in their interoperability characteristics or isolated, so that future replacement is easier and less disruptive.

- Ongoing enhancements to the installed base that exploit new technology offerings or make the existing system more productive or effective to the business and technology subarchitectures will be implemented.

1.2.3 GOALS OF THIS DOCUMENT

The definition and description of an EISA, especially for a large organization, is a resource-intensive activity. The goals of this document are to:

- Provide guidelines for applying the TISAF methodology
- Provide systematic process for developing an EISA
- Provide techniques for developing architectural views
- Provide templates for EISA descriptive documents

1.3 TISAF OVERVIEW

The TISAF describes an architectural framework that can support the Department's business processes as they are redesigned to meet the requirements of recent legislative and regulatory changes as well as responding to the rapidly changing technology environment. Departmental architectures must be closely aligned to the most pressing and critical strategic and operational business needs of the individual organizations. The Department sets forth the policies, standards, and guidelines necessary to design, develop, deploy, and support these architectures within the Department's components.

The purposes of this framework were to provide

- A common understanding of the Treasury information technology vision
- A unifying concept, common principles, common terminologies, and common standards for Treasury information systems
- A context for identifying and resolving policy, management, and strategic technical issues
- A context for strategic planning and budget formulation of Treasury information systems
- A template for the development of enterprise information system architectures

The major elements of the TISAF are described briefly in the following sections.

1.3.1 TREASURY ARCHITECTURE VISION

The Treasury architecture vision supports the Department's business vision. It involves

- Establishing a comprehensive, enterprise-wide architecture that supports the Department's IRM mission and allows achievement of the DASIS mission goals and objectives
- Use of appropriate business process methodologies to create and manage the links between departmental mission needs and the ISAs
- Maximizing the use of information and applied technology to enhance productivity, support business processes and meet the challenges of the Department's mission

1.3.2 ENTERPRISE INFORMATION SYSTEM ARCHITECTURE

The Treasury EISA incorporates three architectural views and an infrastructure:

- A *work architecture* that specifies the decentralization of the business, the distribution of the work organizations to business locations, and the communication and coordination between these locations. It also describes the major operations performed by work

organizations in support of functions and the types of work in terms of the type of workers and types of work location.

- An *information architecture* that identifies, defines, and organizes all of the information needed to perform the enterprise business operations and the relationships among that information. All data needed to support business functions should be captured in the information architecture.

- A *functional architecture* that identifies, defines, and organizes the business functions, processes, or activities that capture, manipulate, and manage the business information to support business operations. It also describes the logical dependencies and relationships among business functions

- An *infrastructure* that specifies the hardware, software, and telecommunications components, management tools, security services, and distributed computing services to support the functional and information architectures.

1.3.3 SUMMARY OF ARCHITECTURAL PRINCIPLES

The TISAF established the architectural principles for the EISA. Tables 1-1 through 1-5 summarize the architectural principles from the TISAF necessary to understand the guidance provided in this document.

1.3.4 TECHNICAL REFERENCE MODEL

The Technical Reference Model (TRM) of the TISAF provides a generally accepted representation of the generic components of an information system. It allows designers, developers, and users to have a common understanding of the services to be provided by the infrastructure. It assists in identifying and resolving issues affecting interoperability, portability, and scalability.

The objective of the TRM is to provide a standardized information system structure that can be used to guide the design and development of customized information systems that meet the specific business needs of the Department of the Treasury bureaus. It describes a set of services that are categorized by functional area. These services may be implemented on a single platform or on a collection of homogeneous or heterogeneous platforms.

Table 1-1. General Architectural Principles

Principle	Description
AF-1: Responsible Information Processing	Information processing activities will be conducted in a responsible manner, complying with applicable laws, orders, and regulations.
AF-2: Common Business Process	Common business process will be implemented in a consistent fashion to provide interoperability and reusability.
AF-3: Open System Standards	A coherent, consistent set of open system standards should be adopted to facilitate interoperability.
AF-4: Systematic Standards Life Cycle	A systematic life cycle will be used to select, review, and retire standards and ensure that adopted standards will be useful to the Treasury community.
AF-5: System Life-Cycle Methodology	Information system projects should adopt a system life-cycle methodology consistent with the requirements, risks, costs, and complexity of the project.
AF-6 Information System Architecture Framework	The information system architecture framework shall be used to guide the design of the EISA in accordance with Treasury goals and objectives.
AF-7: Department-wide Information Technology Policy	Information technology policy will be coordinated Department-wide, but implemented by each bureau.
AF-8: Department-wide Information Systems Security Policy	All Departmental information systems should conform to the Departmental information system security policy.
AF-9: Information Resources Protection	The Department should protect information resources commensurate with the risk and magnitude of harm that would result from their loss, misuse, or unauthorized access or modification.
AF-10: Protection of Individual Privacy	The Department should protect the privacy of information about individuals.
AF-11: Multidisciplinary Security Mechanisms	The Department should develop security mechanisms based on a multidisciplinary approach.
AF-12: Measurement of Applications and Infrastructure	Applications and infrastructure components will be designed and implemented to facilitate monitoring and measurement.

Table 1-2. Work Architecture Principles

Principle	Description
WA-1: Information Independent of Physical Location	The Department will provide access to information to authorized users to perform their jobs independent of their physical location.
WA-2: Information Capture Close to Source	Information should be captured in computer-readable form as close to the source of origin as possible, including external sources and forms prepared and submitted by the public.
WA-3: Common User Interfaces	Common user interface components and standards should be used to provide user interface services.

Table 1-3. Information Architecture Principles

Principle	Description
IA-1: Data and Information Sharing Standards	Data and information sharing standards will be implemented Department-wide.
IA-2: Department Wide Data Management	Data is a vital corporate asset which should be managed to ensure its quality, timeliness, security, and availability Department-wide.
IA-3: Security of Corporate Information	The security of all corporate information will be ensured.

Table 1-4. Functional Architecture Principles

Principle	Description
FA-1: Standards-Based Services	Standards-based services required by applications will be obtained from the infrastructure.
FA-2: Distributed Applications	An application can be distributed across multiple platforms.
FA-3: Application Sharing of Computer Resources	The physical resources of a computer system may be shared—either simultaneously or sequentially—among different applications.

Table 1-5. Infrastructure Principles

Principle	Description
IN-1: Standard-Compliant System Components	Information systems will be designed and implemented using standard-compliant system components.
IN-2: Interoperable Corporate Computing Resources	Corporate computing resources will be interconnected to promote interoperability consistent with information technology policy.
IN-3: Modular Components with Standardized Interfaces	The infrastructure will use modular components with standardized interfaces to support flexibility, scalability, reusability and evolution.
IN-4: Information System Architecture Framework Compliance	Information system architectures should be designed to comply with the information system architecture framework.
IN-5: Multiple Tiers of Functionality	Information system architectures will have multiple tiers of functionality.
IN-6: Common Services	A common environment that provides for the delivery of common services to all subscribers will be implemented.
IN-7: Network Access	All workstations will be attachable to the network with secure communication linkages to all required servers.

The TRM consists of the following basic entities:

- Application software and tools
- Application programming interface (API)
- Application platform
- External environment interface (EEI)
- External environment

The application platform provides the six categories of services identified by the TRM:

- Application services
- System services
- Communication services
- Management system services
- Information security services
- Distributed computing services

More detail regarding the nature of these services can be found in section 4 of the TISAF.

1.3.5 TARGET TREASURY STANDARDS PROFILE

The TISAF describes a target standards profile for use by the Treasury bureaus. This profile describes both mandatory standards, which must be used by the bureaus, and optional standards, which are suggested but not required for use. However, the ultimate goal is to migrate more of the optional standards—as they become accepted by the bureaus and by industry—into the mandatory category.

The candidate set of Departmental information technology standards represents guidance for achieving higher degrees of interoperability within the Treasury community and with other business partners and stakeholders. It is strongly recommended that new contracts be compliant with these candidate standards. The Department will measure selective aspects of ISA implementations to assess the extent to which these contracts measurably contribute to the Department's investment returns.

As higher degrees of interoperability are deemed critical, especially within the business areas and across administrative functions, the need for standards becomes critical. Without the adoption of common standards as guidance, achieving higher degrees of interoperability throughout the Department is unlikely. Standards are and continue to be an important pillar in the Department's architectural framework and they will be advocated and stewarded by the CIO for the benefit of the Department.

1.4 DOCUMENT ORGANIZATION

This document is organized into three sections and four appendixes.

Section 1 is the introduction to this document. It presents the background for the creation of this document, provides a brief overview of the TISAF, and a describes the applicability of this document to the Treasury Department.

Section 2 reviews the components of an EISA as presented in the TISAF, describes some of the techniques that can be used to develop each of the architectural views, and provides a mechanism for assessing an architecture.

Section 3 presents the architecture development process, a methodology for developing and updating information system architectures and planning for their transition and implementation.

Appendix A provides a definition of the acronyms and phrases used throughout the document.

Appendix B provides a set of references for concepts and information contained within the document.

Appendix C provide the templates and outlines of the architectural documents to be produced during the architectural development process presented in section 3.

Appendix D describes the mechanism for proposing changes to this document.

SECTION 2
EISA DESCRIPTION

The definition of an information system architecture has become recognized as critical to large-scale system development efforts. An architecture, by providing explicit models, enables the identification and reasoning of enterprise-level and system-level concerns. These concerns are critical when attempting to develop an integrated set of systems that, as a whole, meet the business needs of an organization. While information system architecture can be thought of as a form of design, it is at the same time distinct from traditional software design (Garlan 1995). Traditional software design does not address the system-level issues in an explicit manner. These include large-scale reusable components and reusable design patterns, and system-wide attributes such as end-to-end performance, maintainability, interoperability and others.

This section starts with a discussion of what an EISA is, its purpose, benefits, characteristics and structure. Next, the various architectures, which model the different perspectives of the overall architecture, are introduced. A list of modeling techniques that are commonly used to define and represent each of the architectures is presented. The relationships between the different architectures is also discussed. Finally, critical issues and concepts in architectural development are reviewed.

2.1 ENTERPRISE INFORMATION SYSTEM ARCHITECTURE

An *information system architecture* (ISA) is the high-level conception of an enterprise in its environment. More specifically, according to the TISAF:

> An ISA is a conceptual and coherent blueprint that describes the structure of information system components, their interrelationships, and the architectural principles and guidance governing their design and evolution over time in an organization.

An *enterprise information system architecture* (EISA) represents an ISA for an entire enterprise or organization. In this sense, an EISA is a special form of an ISA.

An enterprise can be made of multiple *business operations*. A business operation is a set of business processes performed to support the business mission. Examples of business operations include investment management and loan processing in a bank as well as billing and customer care in a telecommunication firm. Each business operation's high-level information system needs are modeled in the EISA and then in more detailed ISAs.

An EISA, then, can contain multiple ISAs, each modeling the detailed information architecture of individual business operations within the enterprise. This concept is discussed in more detail later in this chapter.

A *business process* is an integrated set of functions that are carried out in response to a request made of a business operation. A business process is associated with a business operation and is

contained with it. A business operation may have multiple processes. The business processes help identify and model the detailed responsibilities and boundaries of a business operation, and identify the functions that are contained in the business operations.

A business process should not be confused with a functional area, which is a collection of functions related for the purpose of defining a single information system. A business process is horizontal view that may cross multiple functional areas and therefore information systems.

An *information system* is an arrangement of functions, information, work organizations, and technologies that are integrated for the purpose of supporting and improving the day-to-day operations in an enterprise, as well as fulfilling the problem-solving and decision-making information needs of enterprise personnel. Each ISA can contain one or more information systems.

An EISA, then, includes the structure and behavior of the entire enterprise in the context of its environment and shows how the enterprise will meet its requirements. The structure is described through a commonly accepted set of components that are the building blocks of the information architecture. These building blocks and their attributes are defined at an appropriate level of detail to support the planning and design decisions that are being made by the system architects.

2.1.1 THE PURPOSE OF THE EISA

In today's world, organizations need to have efficient, flexible, and streamlined business operations in order to meet the demands placed on them. Information systems need to be structured to help the organization make maximum use of its resources, respond as quickly as possible to customer demands and outside pressures, and adjust to changing conditions.

An EISA is a model of how information technology will support these goals, and it allows critical thinking and explicit representation of how integrated sets of information systems should be structured to implement these goals. An EISA describes how information systems, applications, and people work together across an organization in a unified and integrated manner. Successful execution of an enterprise's business operations requires that the EISA address three critical architectural views: functional, work, and information. These views are discussed more fully later in this chapter.

An EISA is not traditionally what is thought of as design. Rather, an EISA has multiple roles and acts to

- Model the specification and organization of a set of integrated components—hardware, software and telecommunications—to be implemented across the enterprise over time

- Specify the functional allocation of requirements to system components, allowing linkages between the business missions, objectives, and requirements to the information systems architecture and resulting information systems

- Serve as the basis for planning and scheduling the development of the information system and subsystems across the enterprise

- Provide an overall blueprint for the individual systems and subsystems designs comprising it

- Provide a framework for rapid modification of information system to address changing business conditions

A consequence of this definition is that every information system has an architecture, whether or not it is written down (Emery, Hilliard, and Rice 1996). Another consequence is that an EISA can guide the development of many information systems.

2.1.2 STAKEHOLDER VIEWS OF AN EISA

An architecture has different meanings and purposes for the different stakeholders who use it. Because of this, an architecture has to be developed with multiple stakeholders in mind. For example, the architecture must be understandable to users and specific enough to enable the developers to generate a system design, yet the architectural effort must be done in a manner that meets the customer's tight time and budget constraints.

So, when developing an architecture, it is important to keep in mind the various stakeholders who will use it. The stakeholders must be identified and engaged in the architectural effort as early as possible to solicit and incorporate their feedback into the architecture. Only with active participation of all the stakeholders can an architectural effort be successful. However, the payoffs for the stakeholders are great. For example, the customer paying for the information systems development efforts can use the architecture to help estimate schedule and budget, since the architecture makes explicit the dependencies between the systems as well as system development priorities. Technical risks that have significant monetary and time costs are also outlined for the customer. On the other hand, a system developer uses the architecture as a blueprint that shows key interfaces, functions, and technologies that will drive the design of the resulting information systems. Table 2-1 summarizes how different stakeholders use the architecture.

2.1.3 ARCHITECTURAL CHARACTERISTICS AND EVALUATION

As mentioned above, an architecture serves many purposes and is viewed by many different audiences. It is important that an EISA provide a clear representation explicitly supporting these purposes, in a manner that is understandable to the stakeholders using it. The architecture needs to be analyzed and evaluated in order to determine whether it is ambiguous, incomplete or inconsistent, etc. It must also be analyzed to determine whether it contains the necessary quality attributes and accurately represents the organization's needs. For example, does it completely specify the set of functionality that needs be implemented across the organization to meet its objectives? Does it allow for the planning and execution of the information systems development strategy? Are processes defined from an enterprise perspective?

Table 2-1. Stakeholder Views of Architecture

Stakeholder	Reasons for Using an Architecture
Customer	• Estimating schedule and budget • Assessing feasibility and risk of the architecture • Tracing the requirements • Tracking the progress of migration and implementation
User	• Determining consistency with requirements and usage scenarios • Accommodating future capacity and growth requirements • Validating performance, reliability, and interoperability
System architect	• Tracing requirements • Supporting trade-off analyses for technology selection • Assessing completeness, consistency, and coherency of architecture
System developer	• Provides sufficient detail for system design • Provides a reference for selecting and assembling technology components • Provides a mechanism for assessing and maintaining interoperability with existing systems
System maintainer	• Provides guidance for system and software modification • Provides guidance for architecture evolution • Provides a mechanism for assessing and maintaining interoperability with existing systems

When developing, reviewing, evaluating, and validating an architecture, keep in mind that it should

- *Be understandable*. An architecture should be developed in such a way that it is readily understandable by the various stakeholders who use it, and enables clear communication between the stakeholders (for example, between the customers, architects, and developers). The architecture should be developed with the stakeholders who will use it clearly in mind.

  Another way that an EISA promotes understandability is through its ability to manage complexity. Since an EISA models an organization's information systems needs, the extent of what is being modeled can be very large and complex. The architecture needs to overcome this problem of complexity. It does so by using modeling and representation approaches that allow for partitioning and decomposition from multiple perspectives. Also, the architectural representation should exhibit loose coupling among components and strong cohesion within the components. Section 2.3 provides more details on modeling techniques used with the different architectures (work, functional, information, and infrastructure).

- *Be complete and consistent*. The concepts that an architecture represents should be stated in such a way that they are interpreted in the same way by multiple users. Although an EISA provides multiple views of the architectures (see sections 2.1.5 and 2.2), there should be consistency as to the meaning of concepts. A modeling approach with a formal

notation, a well-defined syntax, and set of rules provides this capability. This characteristic will sometimes conflict with the need to be understandable, since many users will not have the experience needed to evaluate a very rigorous or formal notation. It may be necessary to represent a concept in multiple ways for the different groups or to relax the formality constraint. A high-level architectural representation such as an EISA does not have to be extremely detailed, but representations in the descriptions should be clear and concise.

- *Be traceable.* The architecture will need to provide traceability to the business drivers, such as requirements and objectives, as well as maintain traceability with the systems implemented from it.

- *Be scalable and adaptable.* An architecture is not a static description. It will grow, evolve and change as the organization's objectives change and technology evolves. The architecture needs to use techniques and tools that support these evolving changes. The use of industry standard modeling approaches and good tool support are important for achieving these characteristics.

- *Be integrated.* Although there are four different views of the architecture, these views need to be integrated to explicitly describe the relationships between components across the views. The perspective allows to consideration distributed computing. For example, for a work location defined in the work architecture, what data entities, modeled in the information architecture are used at that location? What functions, defined in the functional architecture are run on what hardware platforms that are defined in the infrastructure?

- *Meet specifications of architectural-level quality attributes.* When an architecture is developed, obviously it must meet the requirements specified for it. One type of requirement, commonly referred to as nonbehavioral or quality attributes, defines certain global characteristics of an architecture. As business operations, processes, and the supporting systems grow in size and complexity, the achievement of quality attributes rests in architectural decisions (Kazman and Bass 1994). These characteristics include performance, portability, and maintainability. A more detailed list is presented below.

Although quality attributes are traditionally thought of as design-level characteristics, these attributes also need to be addressed at the architectural level. These attributes can have a profound effect on the manner in which the resulting information systems will be implemented. The architecture explicitly models the nonfunctional qualities to be achieved across systems. In fact, for large complex information system development, the architecture is critical to explicitly defining and achieving these attributes. These enterprise qualities will then be allocated down to the architectures for individual information system(s).

The architecture provides the framework to define and model the quality attributes, allowing developers to define a cohesive and comprehensive vision of the information system landscape, before stovepipe development makes an integrated enterprise development approach impossible.

For example, a single business process may cross many functions and therefore utilize many information systems and applications. However, the process itself has quality attributes associated

with it as a whole (e.g., how fast it needs to execute, how reliable it has to be, what its security requirements are). Only if these attributes are defined at an architectural level can they be properly allocated and further defined at the systems level. For example in figure 2-1, performance for a business process, which uses multiple information systems, is allocated across those systems.

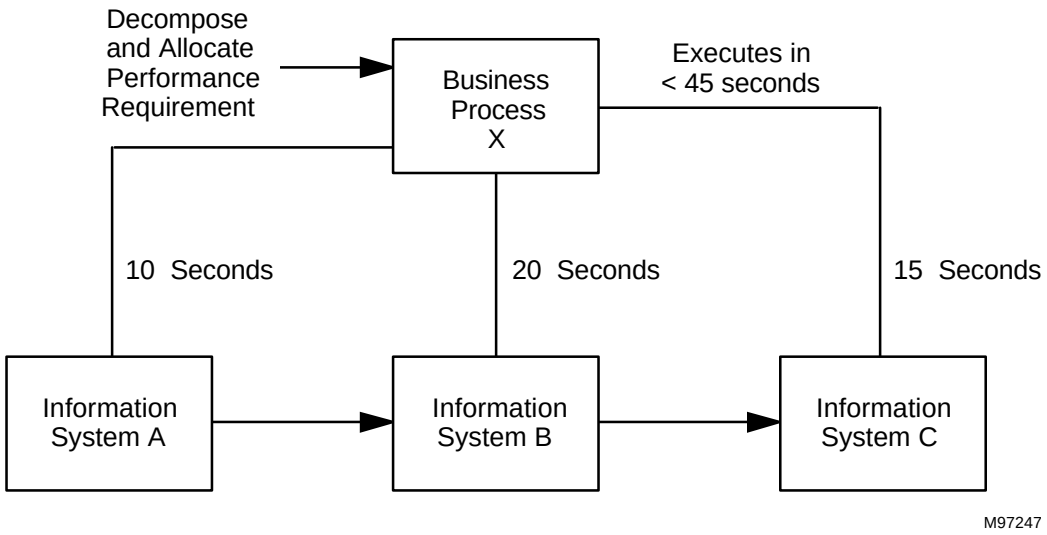

Figure 2-1. Performance Allocation

The quality attributes also act as key criteria used to evolve an architecture. The values of these attributes are defined by the business or organizational needs as well as technology considerations. These attributes include:

- *End-to-end performance.* Business processes within an organization can have performance constraints. These include response time for specific functions or processes.

- *Reliability.* Certain processes have to be architected and implemented in a manner that assures specific reliability. Reliability is the ability of the processes to behave consistently in a user-acceptable manner when subjected to an environment in which it was intended to be used.

- *Availability.* Closely related to reliability, this attribute specifies when a process or operation needs to be available to the user.

- *Security.* Security is the degree to which information access is protected within the process.

- *Maintainability.* Maintainability is the degree to which the components of the process can be easily maintained and evolved.

- *Portability.* Portability is the degree in which software, supporting the processing and running on one platform, can be converted to run on a different platform.

- *Interoperability.* Interoperability addresses how the different pieces of the architecture—software applications, hardware platforms, networks, etc.—will work together.

A key issue is how the quality attributes impose constraints on the architecture and how they can be traced through implementation and testing. Also, there is no correct generic value for each of these attributes. Rather, business requirements, technical feasibility, cost, resources, and development time all play a role. Also, the attributes can conflict with each other. For example, very high performance sometimes may only be achieved at the cost of some maintainability.

So, when an architecture is developed, saying "portability is important" is not sufficient. First, those quality attributes of interest must be determined and then prioritized. Quality attributes do not exist in isolation, but only have meaning within a context. These decisions must be made in terms of business objectives, requirements and anticipated usage.

2.1.4 ARCHITECTURAL EVALUATION AND VALIDATION TECHNIQUES

Because of its importance, an architecture will need to undergo regular evaluation and validation, to ensure that it meets its requirements. It is in an organization's best interest to evaluate the architecture on an ongoing basis to ensure that it both addresses the organization's needs and does so in a quality manner. An architecture evaluation and validation assesses the ability of the architecture to support these aspects. Evaluations address whether the architecture has been developed in a quality manner. Validation addresses, whether the architecture meets the organization's requirements and needs. Evaluation and validation should consider whether the architecture contains the characteristics outlined above. For example, are there minimal connections between components and is the functionality defined in the components encapsulated?

Various techniques can be used to evaluate and validate an architecture. Each of these techniques uses a different approach and can be used either to cross-validate the others or to validate different aspects of the architecture. A representative sample of evaluation techniques includes checklists, scenario analysis, prototypes and simulations, and traceability matrixes (CMU 1997).

Checklists

A checklist (or questionnaire) is a set of questions that is developed after experience evaluating and validating a common (usually domain-specific) set of architectures. Checklists can contain general and relatively open questions that apply to all architectures and/or questions that focused on particular qualities of the architecture being evaluated. Specific quality metrics can be included with the checklists. Examples of checklist questions that address the architecture include: Are the interfaces across and between the systems defined? Have performance criteria been defined for each business process?

Checklists can both focus on the details of the architecture itself as well as addressing the process used to generate and document the architecture. Examples of questions that address the process include: How will you go from here to develop the target architecture? What techniques are you using to represent the architecture so it is understandable to all stakeholders? How are you ensuring that architecture is adhered to when you develop the systems?

Checklists evolve over time. If an organization is just beginning to perform architecture evaluations and does not have checklists in place, scenarios should be heavy stressed.

Scenario Analysis

Scenario analysis is a means of validating the architecture through the use of descriptive scenarios. Scenarios normally describe an event driven collection of functions or activities. Scenarios provide a means to characterize how well a particular architecture responds to the demands placed on it, within the context of the functionality and requirements defined in the scenarios. Many software quality attributes are too complex and broad to be evaluated on a simple scale. Scenarios provide a descriptive means of specifying and evaluating quality attributes within a context. When creating and organizing scenarios, the perspective of the stakeholders to the architecture can be used for scenario generation.

Scenarios, at least as they are initially defined, tend to be architecture-specific but may be reused on future architectures, depending on the particular issue a scenario is addressing. Even if checklists are being used, scenarios are useful in dealing with the issues not covered in the checklists and can also be used to create checklists for future evaluations.

Prototypes and Simulations

Prototypes are working models (in the case of simulations, dynamic models) of some aspect of the architecture. Prototypes and simulations can be used to understand and assess the architecture, particularly those pieces that do not lend themselves to scenarios and checklist.

When considering the use of prototypes and simulations, keep in mind that they are useful in answering performance and technical feasibility questions. However, developing a prototype to answer questions concerned with modifiability will probably not be useful.

Traceability Matrixes

Traceability matrixes maintain linkages between the drivers for the architecture such as business requirements, mission statements and technical issues papers and the architectural decisions themselves. When validating an architecture, the reasons for each architectural decision should be clear and traceable to some form of documentation.

2.1.5 EISA LEVELS

Just as a traditional design may be at multiple levels of abstraction (e.g., logical design, detailed design), so too can an EISA be applied at various levels of abstraction. An EISA defines multiple levels of architecture, as mentioned earlier. An EISA, which is developed for an entire enterprise, may have multiple ISAs as subarchitectures—one for each business operation, as described in figure 2-2.

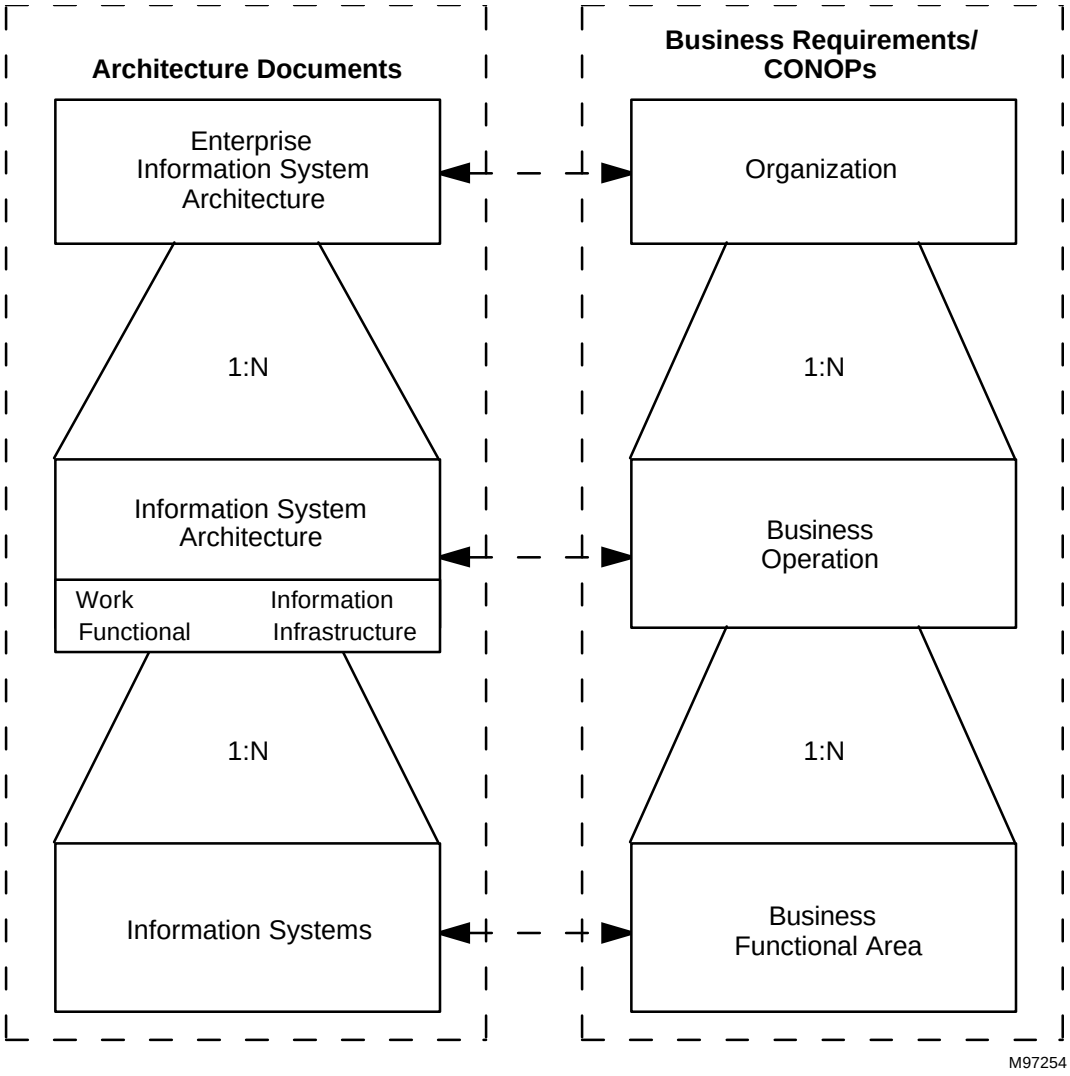

Figure 2-2. EISA levels

This partitioning of the EISA into multiple levels helps to bring focus to

- Those architectural issues that need to be addressed and represented on an enterprise level
- Those more detailed architectural concerns that are best modeled at a lower level of abstraction such as the business operation level

High-level functionality, entities, systems, and interactions between the business operations can be represented at the top level of the EISA. For example, an EISA can model the business operations in an enterprise such as submissions processing, customer service, and payroll as well as the interactions these business areas have between each other. The issues best addressed at an enterprise level include

- Enterprise objectives for information systems
- Information systems planning needs at an enterprise level (e.g., estimation of costs and schedule)
- Modeling and identification of business requirements at an enterprise level, such as requirements that cross multiple business operations
- Determination of common functions that can be reused across an enterprise
- Determination of processes and operations that will cross multiple organizations or functional areas
- Allocation of process requirements across the business operations that make the enterprise
- Enterprise-wide nonbehavioral requirements (e.g., standard languages, maintainability, hardware and system software, portability)
- Modeling of the common infrastructure for the enterprise (e.g., Internet versus intranet)

Each of the business operations can then be individually modeled in more detail by information system architects. Finally, within a business operation, *functional areas*—groups of cohesive functions—may exist. The functionality comprising a functional area can be implemented as an information system or application. For example, in the business operations of a bank's loan processing, functional areas such as credit validation and loan account management exist. Each of these areas can be implemented with an information system or application.

The issues dealt with at the business operation level or lower focus on:

- The more detailed ISA architectures needed at this level
- The concerns specific to an individual business operation information system architecture (ISA)
- The detailed implementation of the functional areas (groups of functions) by a specific information system or application

This breakdown structure for the EISA also supports the sequencing and partitioning of the EISA effort itself. This allows for incremental development of the architecture. For example, the top level architecture can be developed quickly, then the more detailed architectures can be developed on the basis of risk and priority. This increases the opportunity for both rapid and timely results.

The EISA and associated ISAs are also divided horizontally into four architectural components which are discussed in detail later in this chapter.

It should be noted that this is a conceptual description of an EISA. A specific EISA may vary from this structure. For example, a small enterprise may only need one level of abstraction, such as combining the enterprise and business operation levels. A very large enterprise can have ISAs that contain ISAs, effectively having many levels of abstraction. Also, the terminology such as business operations, functional areas and function are meant to represent sets of business functionality at multiple levels of abstraction. An individual enterprise may wish to customize these names and their definitions to better suit the organization.

2.2 EISA ARCHITECTURAL VIEWS

The TISAF describes an enterprise information system architecture in terms of four architectural views:

- A *work architecture* that specifies the decentralization of the business, the distribution of the work organizations to business locations, and the communication and coordination between these locations. It also describes the major operations performed by work organizations in support of functions and the types of work in terms of the type of workers and types of work location.

- An *information architecture* that identifies, defines, and organizes all of the information needed to perform the enterprise business operations and the relationships among that information. All data needed to support business functions should be captured in the information architecture.

- A *functional architecture* that identifies, defines, and organizes the business functions, processes, or activities that capture, manipulate, and manage the business information to support business operations. It also describes the logical dependencies and relationships among business functions.

- An *infrastructure* that specifies the hardware, software, and telecommunications components, management tools, security services, and distributed computing services to support the functional and information architectures.

The relationships of the EISA views are depicted in figure 2-3.

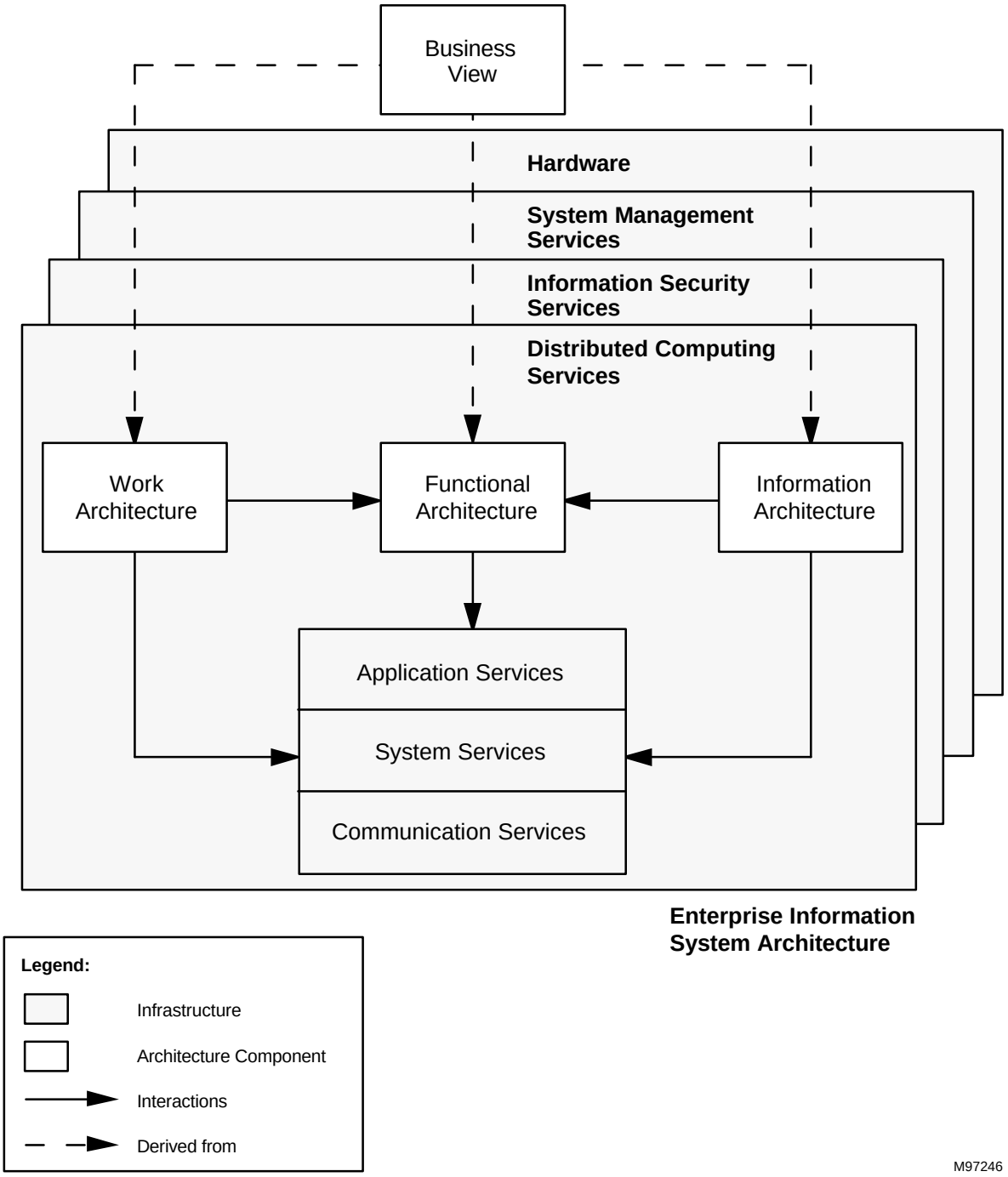

Figure 2-3. EISA Architectural Relationships

2.2.1 EISA LEVELS AND ARCHITECTURAL VIEWS

Each view represents one horizontal dimension of the EISA, with the enterprise level, business operation level, functional area level, function level representing the vertical dimensions. The mapping of the levels against the architectural views is presented in table 2-2.

Table 2-2. EISA Architectures at Each Level

Views/ Architecture Levels	Work View	Functional View	Information View	Infrastructure View
Enterprise level	• Enterprise organization structure • Customers	• Business scenarios • Business objectives and mission	• Common business entities and objects	• Technical reference model Corporate standards • Corporate platforms
Business operation level	• Departments • Work locations • Workflow models • Organizational structure	• Business processes	• Business entities and objects	• Business operation specific platforms and relationships
Functional area level	• Sub departments • Key personnel	• Information systems	• Logical data models	• Information systems specific platforms
Function level	• Users • User locations	• Information system functions	• Physical data models • Databases	

This table outlines the representations that occur in each architectural view, at different levels of the EISA. In addition to the concepts discussed in the beginning of this chapter, key representations and their descriptions include:

- A *business scenario* is an integrated set of business processes that can cross multiple business operations. These scenarios normally describe an enterprise response to an external event.

  These scenarios help to identify and model those activities that are performed across business operations and use multiple business processes. They provide an enterprise level view of how the business performs its work. They also help to define the responsibilities and boundaries of the business operations as well as the business processes needed in each business operation.

  The scenarios can be modeled very formally with such techniques as functional data flows, object interaction diagrams, or use cases. They can also be modeled informally with text descriptions or high-level graphics.

- *Common business entities and objects* are major information entities and their relationships, which are critical for the enterprise and/or shared across the enterprise. They allow for the definition of information so it can used and shared by the business operations and processes.

The common business entities and objects are not meant be an exhaustive data model of the enterprise; rather, they are a high-level model of the most important information and its relationships. This information can be found in a number of ways, including review of the business scenarios for information that is passed by or used by multiple business processes. These entities can be modeled using any standard data or information modeling technique.

- A *department* is a major organizational unit of the enterprise that carries out business processes. A department is a physical concept. A business operation may be associated with one or more departments and vice-versa. However, there is normally a strong correlation between a business operation and an individual department

- *Business entities and objects* are information entities and their relationships which are used by a particular business operation and/or shared across the business processes within that business operation. They allow the definition of information so it can used and shared by the business processes and functions.

2.2.2 INTERDEPENDENCIES AND INTEGRATION OF THE ARCHITECTURAL VIEWS

The four individual architectures are not developed in isolation from one another; rather, each architecture imposes dependencies on and interacts with, the other architectures. For example, in order to understand and bound what information a enterprise requires, the functions of the business need to be specified. The information specified in the information architecture will be used by, and will influence the development of, the functional architecture. And vice-versa, the definition of the functional architecture can motivate the need for new information and information relationships. Because of this, it is extremely important that the architectures not be developed in a stovepipe manner, by separate teams that have little interaction with one another. The architectural development process should be an integrated effort. Figure 2-4 depicts an expanded model of an information system architecture showing the relationships with the architectural views overlaid on it.

The linkages between the work, functional, information and infrastructure views need to be defined explicitly and maintained as each view is developed. Techniques and tools used include mapping tables, such as CRUD matrixes, scenarios, checklists, and integrated repositories with consistency rules and reporting capabilities.

2.2.3 EISA AND REUSE

An EISA also provides a generic blueprint and guidance that can be reused by multiple information systems. For example, suppose an enterprise has a business operation for collecting information. There may be three ways to collect that information: voice, electronic, and paper submissions. Each medium of collection has unique requirements, but all essentially share a common architecture. By providing a common architecture, the EISA facilitates the reuse of both knowledge as well as components across the various business operations, as in the above example for collecting information. This reduces the need to rebuild or reinvent a solution that already exists, rather, developers can draw on proven approaches. In this way an EISA acts as a domain

model which is a conceptual representation of the entities and processes within a problem domain, such as avionics, medicine, or law enforcement.

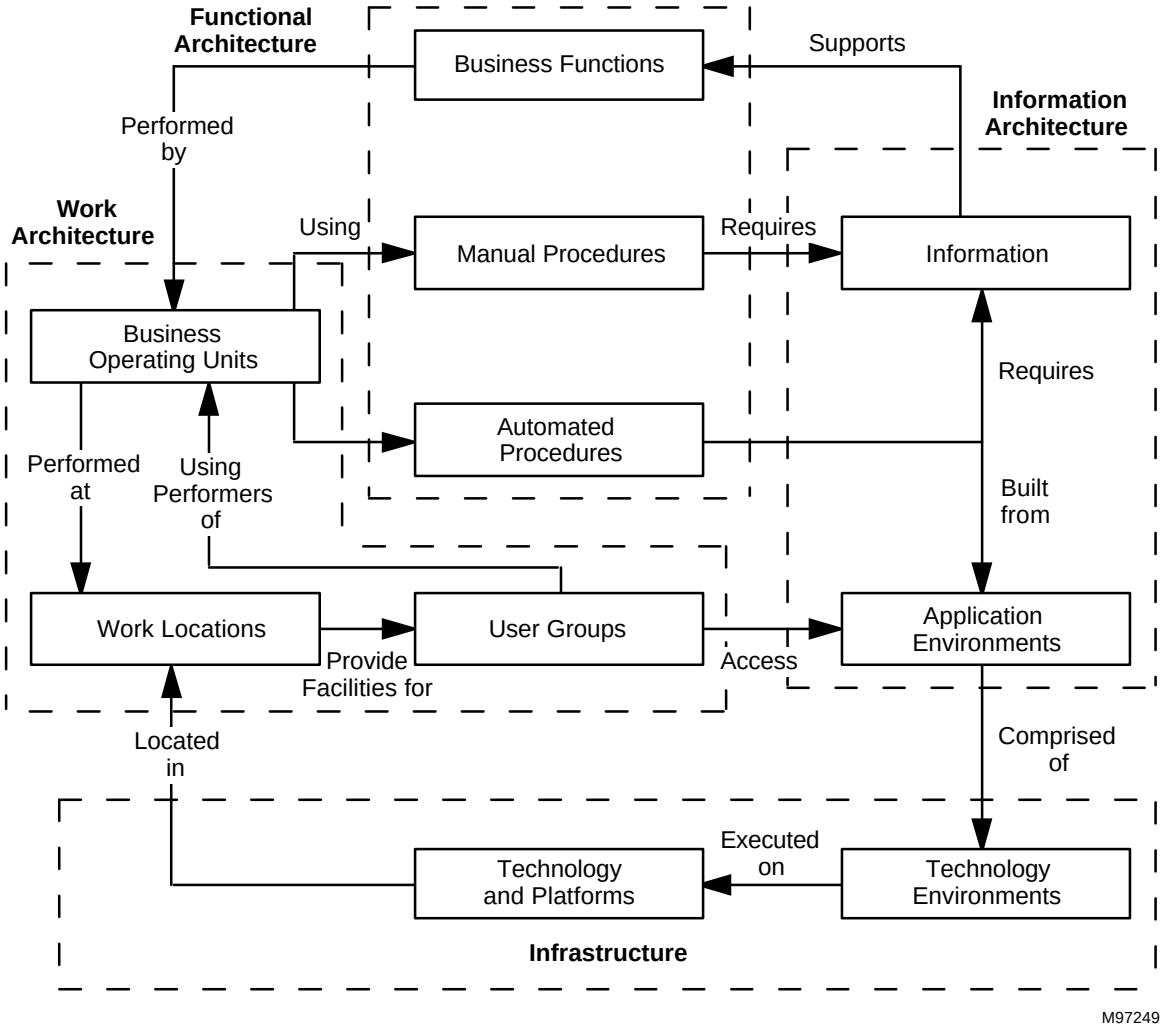

Figure 2-4. An Integrated Model of an ISA

2.3 COMMON TECHNIQUES TO MODEL THE EISA VIEWS

A number of techniques and approaches are used to represent and model the different architectures. A high-level description of the most common approaches, methods and techniques is presented below with each architecture. This list is not intended to be exhaustive; rather, it is intended to give a flavor of the techniques available. Also, any technique should be supplemented with supporting text and tables to allow for additional description of the architectural concepts. In some cases, text alone may be sufficient to describe some aspects of an architecture.

2.3.1 GUIDELINES FOR SELECTION OF TECHNIQUES

There are many techniques and approaches that can be drawn upon when developing an architecture. No one technique is always "right" or always "wrong." Rather, in one situation a certain technique may be appropriate because of certain conditions. Any acceptable means for modeling business processes and information flows can be used. A particular structured or semistructured analysis approach can use a family of techniques to analyze, evaluate, develop, and deploy a system design within an architectural framework. The use of models helps to ensure that a quality approach is taken in architecture evolution so that these processes can be repeated, that single design decisions can be reused elsewhere as necessary, and that a basis exists for managing changes and describing investment alternatives. For example, the use of simulation models for large, complex EISAs is very useful for early end-to-end performance analysis.

The specific set of techniques for a given architectural effort is left to the discretion of the organization on the basis of its ability to achieve the Department's architecture principles and objectives. However, developing and documenting architecture designs with formal techniques is preferable to merely intuitive or ad-hoc alternatives. When selecting techniques for an architecture, keep in mind that they should:

Support the development of the different architectures. The technique being used should be appropriate for the architecture that it is modeling. For example, data modeling techniques are very useful for representing the information architecture, but are not good at representing the functional architecture. This means that most architectures will utilize multiple techniques.

Have a coherent modeling language. An architectural technique ideally should have a simple, clean consistent structure with distinct semantic concepts. The syntax and semantics of a technique should be relatively easy for users to grasp, yet powerful and robust.

Be teachable. The technique should be designed to be taught to and used by business professionals and system developers as well as system architects. Thus, it can serve as an effective communication tool across interdisciplinary teams.

Be well-tested and proven, yet flexible. Ideally, a technique should have a track record on a number of development efforts. It should be thoroughly tested and shown to produce repeatable results. At the same time, every organization and effort is different, so the technique should allow for its own modification to adjust to these factors.

Be automatable. The technique should be supported by a variety of graphics packages. Commercial software should exist that supports the refinement, analysis, and configuration management of models developed with the technique.

The sophistication of techniques used should be based on the level of prior experience with architectural modeling, the size, and the strategic importance of the effort as well as prior investments in place. Other factors to consider include the ability to reuse analytical products, meta-data, data structures, and generated code.

When selecting a technique or set of techniques to use, keep in mind that (Davis 1993):

- For any specific architectural effort, the skills of the architects and the user's needs should be considered before selecting a technique.
- Given any specific architecture effort and the skills of architects and users, there still may be more than one ideal approach.
- Organizational approaches and standards need to be factored into the selection of a technique.
- No single approach will suffice in a complex, diverse environment. Knowledge of a specific technique is very useful, however, you should have a toolkit of techniques to draw from.
- Knowing when to apply a specific technique is as important as knowing how to apply it.

Below are some additional attributes of good architectural techniques. Not all techniques will meet these attributes, and some of the attributes may conflict with each other.

- The technique should reduce ambiguity and make explicit the key architectural issues and concepts.
- The technique should be understandable to its audience, which normally includes customers, users, architects and designers.
- The technique should promote reuse by being understandable and providing a structured approaches and notation.
- The technique should support easy modification.
- The technique should permit annotation and traceability.
- The technique should allow the partition of the problem is manageable pieces.

The four different architectural views of an EISA are briefly described below, along with a list of techniques that can be used to develop them. For the selected set of techniques, more detail is provided in the *Treasury Architecture Development Guidance*, appendix C.

2.3.2 WORK ARCHITECTURE

The purpose of a work architecture is to portray the major current work processes and activities performed by individuals within the work environment. The work architecture identifies the roles, processes, and activities employees will perform when the architecture is implemented. The work architecture also documents work locations and organizational hierarchy.

2.3.2.1 Key Aspects of Work Architecture Development

The work architecture

- Defines the various work roles and each roles responsibilities
- Models the high-level work processes, the work roles involved and the rules governing the processes
- Supports the specification of work locations and the communication between them

It should be noted that the emphasis of the architecture development effort is on information system functionality, but work processes and activities need to be identified and explored for discovering needed data and business/information needs.

The distinctions among functions, processes, and activities are that an organization has functions or tasks to perform but in order to do these, processes which are composed of activities are necessary. Processes are step-by-step activities; functions are not step-by-step but a listing of the work. For example, a function could be "pays bills." The process for this could be to "validate the bills, write the checks, and mail the bills."

To develop the work architecture, multiple components of the ISA must be analyzed as depicted in figure 2-5.

By examining the components in the work model, the architectural vision and the resulting information systems can be assessed for how they impact work activities, change skill requirements of users, affect functional operating locations, and eliminate or reduce manual support systems.

The work architecture describes the major operations that are being performed by workgroups in support of a business operation, functional area, process, or activity. The types of work (business operating units) need to be defined and described in terms of the types of workers (user groups) and the types of work locations (work locations).

This model recognizes that "networks" of individuals coexist in an organization to support manual and automated systems. It supports the "team" concept and the multiple roles that individuals may have in the business unit, and recognizes that teams consist of individuals who may work remotely from each other.

To define the work architecture, special consideration needs to be given to concepts being defined in the other architectural components, such as the manual and automated procedures, defined in the functional architecture and the information specified the information architecture.

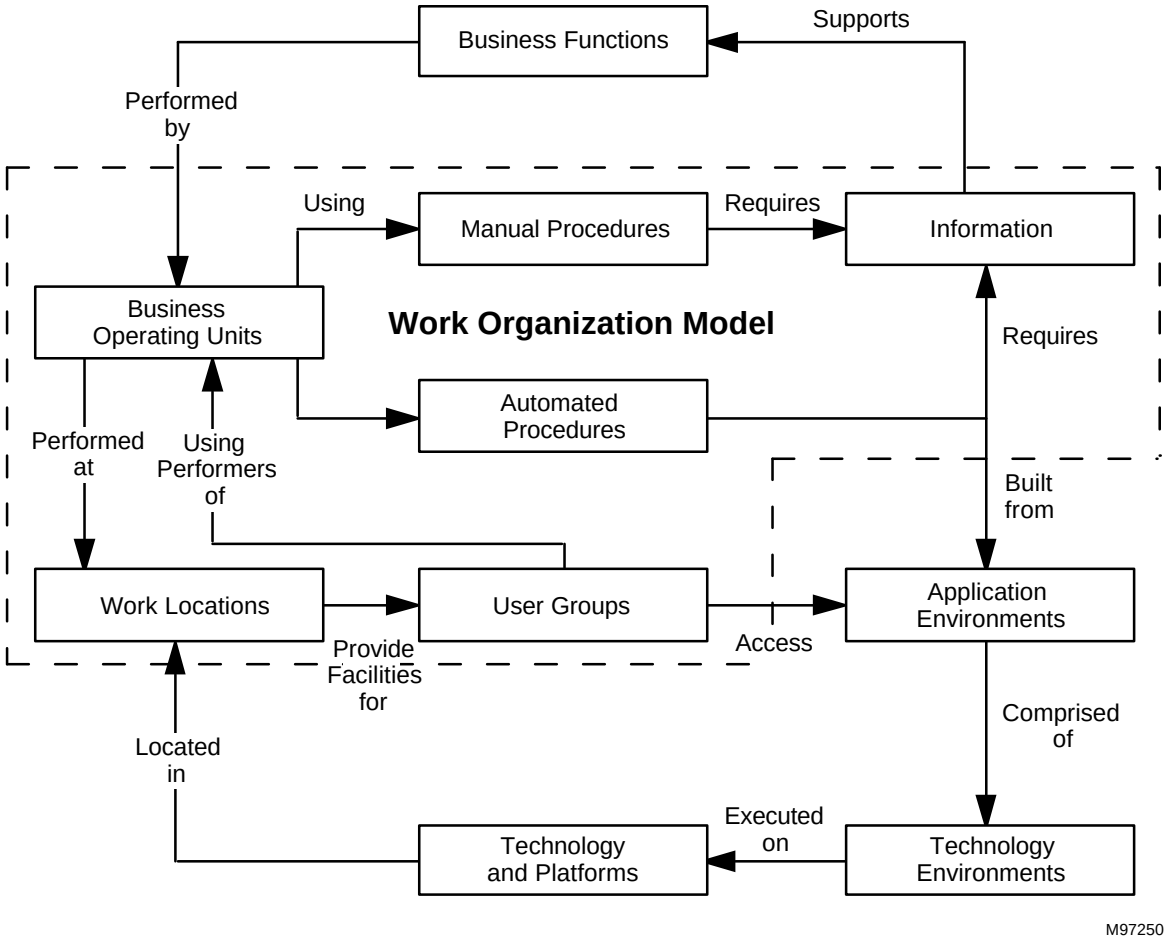

Figure 2-5. The Work Model

2.3.2.2 Possible Methods and Techniques for Work Architecture

Approaches used to represent and model the work architecture include:

- *Organizational structure and hierarchy charts.* These charts describe the structure of a organization including the organizational business units, the individuals within the business units, their role and title, and the location from which they operate. The charts are useful in identifying individuals who have a stake in the ISA effort and who, therefore, will need to be consulted or involved in some way in the effort. They are also helpful in highlighting data and application sharing across organizational units.

- *Workflow Models.* Workflow is about the way in which work and its associated documents (e.g., tax forms, bank loans) move from person to person and from place to place in order to accomplish some business process (e.g., tax return processing). Workflow specifies the value that is added at each step. During a business process, specific activities must occur on a document in a specific order, and if exceptions occur, then what needs to be done to deal with these must be specified. These processes may be very collaborative across the individuals involved in the processes.

Workflow models describe the "what, when, and to what" aspects of the workflow. They need to model parallel routes and have business logic for dynamically determining routes based on normal as well as exception processing. Workflow models specify the routes, rules and roles of an workflow process.

- *Block Diagrams.* These are high-level diagrams, which can be very informal, and represent the major business operations and the primary, high-level interactions between the business operations.

  Block diagrams are easy to understand and develop. They can be used when the stakeholders reviewing the diagrams are diverse and not experienced in the more formal notations. The are also very effective at getting a simple, high points across.

- *Business process reengineering (BPR) techniques.* BPR is an approach to implementing fundamental change in the way work is performed across the organization to achieve improvements in critical measures of performance, such as cost, quality, service and speed. BPR is performed with the entire enterprise in mind, focusing on business processes, not individual organizations or functions.

  BPR should be considered when there is a need to fundamentally change the business operations of an organization. Normally, if a BPR effort is performed, it is done outside the EISA effort and used as input. The BPR models and diagrams, however, can be incorporated within the EISA work architecture.

2.3.3 FUNCTIONAL ARCHITECTURE

The purpose of the functional architecture is to model the functions, processes, and activities of the organization can be supported by information technology applications, thus providing a high-level description of application opportunities and showing logical dependencies among opportunity areas.

2.3.3.1 Key Aspects of Functional Architecture Development

The functional architecture

- Documents the functions that take place within the architecture—normally explicit functions, objects with behaviors, or processes
- Provides a means to decompose the functions into more detailed functions or logic
- Supports the partitioning of the functional architecture
- Allows the viewing of the functional architecture from different perspectives
- Models the interactions between the functions and the processes across functions

- Defines the system(s) boundaries and the information and interactions across the boundaries

To develop the functional architecture, multiple components of the organization must be analyzed as depicted in figure 2-6.

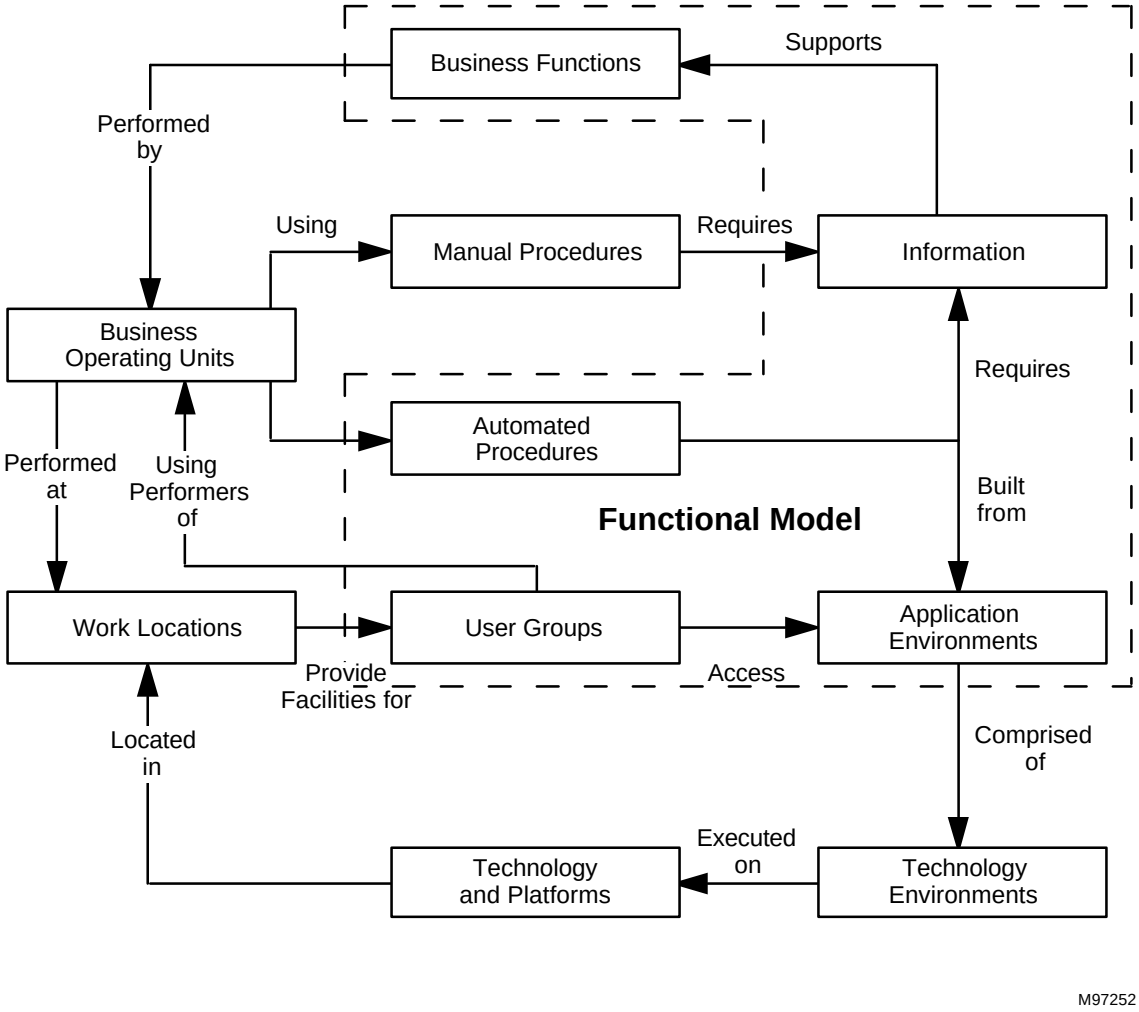

Figure 2-6. The Functional Model

By examining the components in the functional model, the architectural vision and the resulting information systems can be assessed for the types of business functions, processes, and activities which are required to support the business operation(s) and users; how functions, processes, and activities will be grouped and interfaced; and what usage levels are anticipated.

To define the functional architecture, special consideration needs to be given to concepts being defined in the other architectural components, such as the information to be used by the functions, the business operations that the functional architecture will be supporting and the user groups involved.

Note that manual procedures are not included in this model because they are not supported by the information systems. To the extent that manual procedures are to be automated in a future version of the information system, they can be included in the automated procedures.

2.3.3.2 Possible Methods and Techniques for Functional Architecture

The following approaches may be used to represent and model the functional architecture:

- *Object-oriented approach.* An object-oriented approach is one in which key concepts are represented as objects. Objects are defined as representing "an individual, identifiable item, unit, or entity, either real or abstract, with a well-defined role in the problem domain" (Booch 1994). An object contains information and a set of operations that act that information. Objects interact, involve one another's operations, to accomplish some business process. Object-orientation provides a more intuitive modeling approach for architecture then many other techniques, since objects allow a natural correspondence with real world entities and their interactions.

  When considering an object-oriented approach, be aware that

  ◊ Object-orientation is designed to accurately reflect concepts contained in distributed and client/server architectures. If these forms of architectures are being modeled, then an object-oriented approach should be strongly considered.

  ◊ Object models provide a comprehensive and complete model, since many object methodologies provide for representing the static information view, behavioral/functional view and state view in one integrated model

  ◊ Object-oriented modeling approach provides more resistance to change, since objects change infrequently while processes and procedures (top-down) are frequently changed, this providing a more resilient functional architecture.

  ◊ Object models provide for increased reuse do to their ability to model looser coupling and information hiding.

  ◊ If an object-oriented design and construction approach is envisioned, then an object-oriented based architecture provides a more seamless transition to the design.

  Issues to address when considering a object-oriented approach include the following:

  ◊ Whether an object-oriented support infrastructure is available. Object-orientation is a relatively new approach and there is are a limited number of experienced architects with an object background. Object-orientation is a very robust and complete modeling approach and takes time to learn. Given this, a good support approach that includes formal training, methodology and hands-on mentoring will be required for an effort to be successful.

◊ If the current architecture and/or system designs are currently in a non-object model then it may be hard to update and maintain it as an object model.

◊ Architectural object modeling is a relatively new approach, there will be less guidance and lack of examples available to review then with other, more established approaches

- *Data flow and functional-oriented techniques.* These models describe the processes and the flow of information between them in order to accomplish some business activity. These approaches are characterized by the way they model the processes and the flow of information between the processes. A data flow diagram is a graphical technique depicting the information flow, the processes it flows through, an how it is transformed in each process. Each process can than be "exploded" or partitioned into a lower level data flow description which contains more detailed processes and information flows.

  When considering a functional-oriented approach, be aware that

  ◊ Functional models have traditionally been used to model mainframe-centric, batch-oriented architecture. These models should be strongly considered if this type of development is anticipated.

  ◊ Data flows are a very flexible technique and can be used to model processes at any level of abstraction, all the way from high-level business processes to very detailed design flows.

  ◊ Functional models are relatively simple compared to other techniques (such as object modeling) and therefore easy to understand and learn.

  ◊ There is significant industry experience with these techniques and it is easier to get architects with experience with this techniques then with others.

  Issues to address when considering a functional-oriented approach include the following

  ◊ They focus on process and are very weak when modeling the modeling the data aspects of the architecture. Normally, a data model has to be developed along with the functional model. However, the models need to be integrated together. When this occurs, incompatibilities and contradictions will likely be found. The integration of the two models requires a the development of a technique and approach for managing this integration process.

  ◊ Functional oriented approaches are considered to be top-down in almost all cases. If an iterative form of development is being used, consider another technique such as object-orientation.

- *IDEF0.* This is a very specific functional oriented technique used to produce a model which is a structured representation of the activities or processes within the environment

or system. It has many of the characteristics of functional oriented techniques described above. The unique characteristics of IDEF0 include its emphasis on

◊ *Control.* Describing things that effect the flow of information

◊ *Mechanisms.* Which describe how a function will perform, who will perform it and what physical resources are needed to perform the function

If the control and mechanisms aspects are important to explicitly document, and a functional approach is to be used, then consider IDEF0.

- *Information engineering.* An integrated set of tasks and techniques for data modeling, business planning, process modeling, systems design, and systems implementation. Information engineering provides a comprehensive set of techniques that can be used for architectural development. Its strength, however, is its strong data and information modeling components.

- *Scenario analysis.* This technique is a means of identifying and validating functionality through the use of descriptive scenarios. Scenarios describe, typically, in an informal format, a string of functions or activities that occur in response to a business event. Scenarios are extremely useful in discovering functionality as well as validity it with the users and customers. A popular form of scenarios is known as use cases.

  Scenario analysis is a very effective means of eliciting and validating an functional architecture. It are normally used in conjunction with other techniques such as object modeling.

- *State-oriented techniques.* State oriented techniques describe the major states or modes of the entities in the business domain. Describing these states and the events that cause them to change, helps to better understand the behavior of the business.

  State oriented techniques are usually used to supplement other techniques such as object-oriented or functional approaches.

- *Simulation techniques.* These techniques provide a means to dynamically model the processes, activities and functions defined in the functional (as well work architectures). These techniques are very useful in the early definition and validation of quality requirements such as end to end performance and capability.

2.3.4 INFORMATION ARCHITECTURE

The purpose of the information architecture is to model the logical view of the information needed by the business areas, functions, processes, or activities of the organization. Moreover, it not only maps out the information needs, but where the information is going and why.

2.3.4.1 Key Aspects of Information Architecture Development

A well-defined information architecture can be

- The means to provide consistent information across all business areas
- The foundation for integrating and organizing information as a company asset
- The fundamental tool for managing the complexity and changing nature of information storage and retrieval technology

To develop the information architecture, multiple components of the organization must be analyzed as depicted in figure 2-7.

By examining the components in the information model, the architectural vision and the resulting information systems can be assessed for what information stores are required to support the functions, processes, and activities; what forms and volumes of information are involved; what relationships between the information stores must be provided; and what access and security controls are required.

The construction of this model requires that all forms of information and their corresponding information stores be included and that due consideration given to the issues of placement and distribution to working locations to support user and application access.

To define the information architecture, special consideration needs to be given to concepts being defined in the other architectural components, such as the procedures and functions defined in the functional architecture, and the business operations and user groups defined in the work architecture.

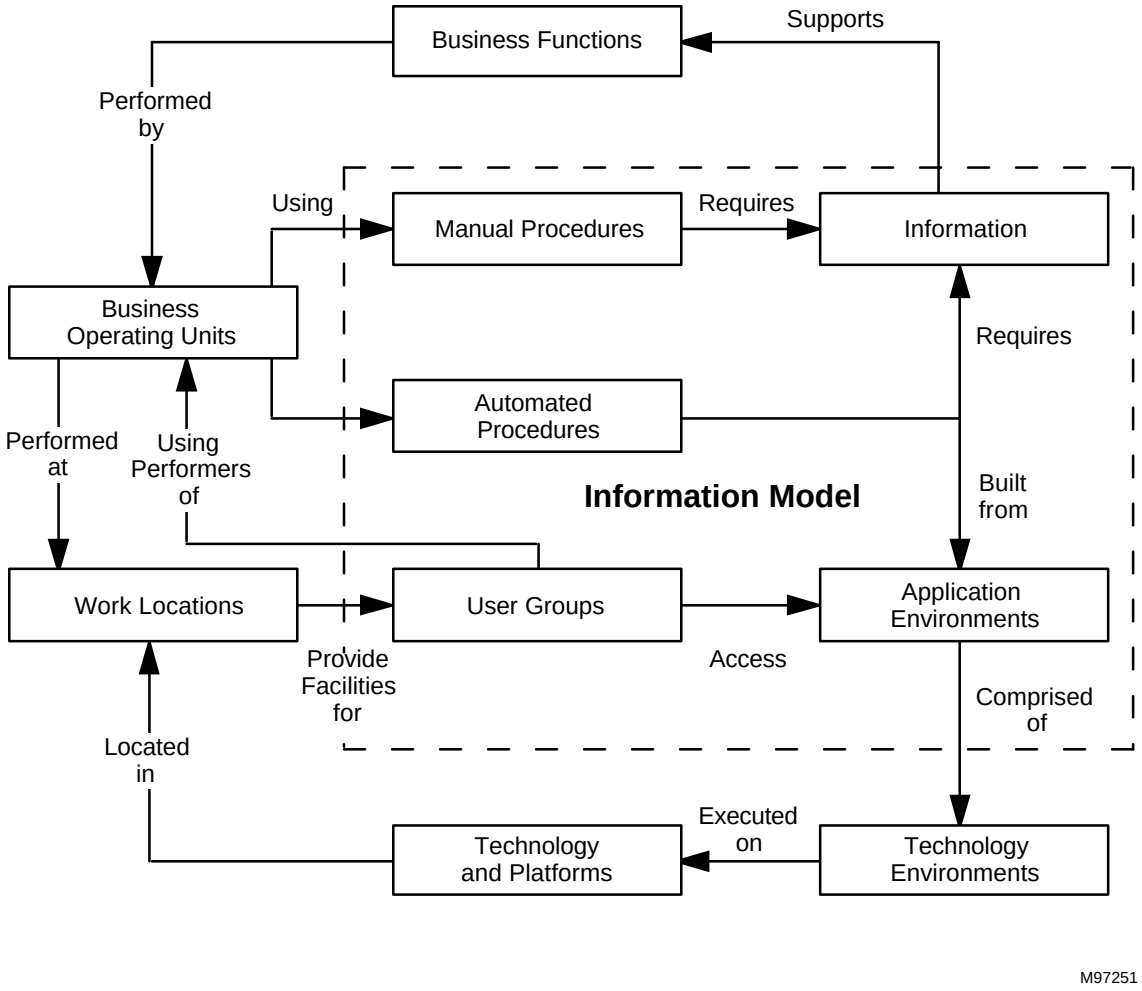

Figure 2-7. The Information Model

2.3.4.2 Possible Methods and Techniques for Information Architecture

The following approaches may be used to represent and model the information architecture:

- *Data modeling.* This technique focuses on identifying and describing the data an organization needs to perform its business functions. Data modeling specifies the information important to the organization as a set of entities as well as the relationships between the entities. Attributes for each entity are also defined.

  When considering a data modeling approach, be aware that

  ◊ Data modeling is normally used in combination with functional techniques such as data flow diagrams.

  ◊ Data modeling is useful for modeling the overall view of the data required to run an enterprise. The models can then be analyzed to identify and scope projects to build shared data resources.

◊ A fully developed model can be used to define an application independent view of data which can be validated by users and then transformed into a physical database design for any of the various DBMS technologies. In addition to generating databases which are consistent and shareable, development costs can be drastically reduced through data modeling.

◊ Since a data model actually represents the infrastructure of an organization, vendor software can be evaluated against a company's data model in order to identify possible inconsistencies between the infrastructure implied by the software and the way the company actually uses the data in business operations and processes.

◊ By defining the contents of existing databases with data models, an integrated data definition can be derived.

Issues to address when considering a data modeling approach include the following:

◊ It should support a consistent model of the business and be true across all application areas.

◊ It should allow the data model to be extendible, such that new data can be defined without altering previously defined data.

◊ It should be transformable both to the required user views and to a variety of data storage and access structures

- *Object modeling.* As mentioned previously, an object-oriented approach is one in which the key concepts are represented as objects. Objects are defined as representing "an individual, identifiable item, unit, or entity, either real or abstract, with a well-defined role in the problem domain" (Booch 1994). Because an object contains both information and a set of responsibilities that act that information. Object models can be used to specify both the information and functional architectures.

- *IDEF1x.* A very specific form of data modeling that was designed with work in conjunction with other IDEF approaches including IDEF0. It incorporates the entity-relationship model.

- *Information engineering.* An integrated set of tasks and techniques for data modeling, business planning, process modeling, systems design, and systems implementation. Information engineering provides a comprehensive set of techniques that can be used for architectural development, its strength, however, is its strong data and information modeling components.

- *Information mapping.* A systematic approach to analyzing, organizing, and presenting information based on a user's needs and the purpose of the information. Information mapping is a technique that emphasizes

  ◊ The user's information needs

  ◊ The purpose, accuracy, and structure of the information

  ◊ The technology the information will be delivered on

  ◊ Organizing and structuring the information on the basis of the way readers learn and read (principles include chunking, labeling, relevance, consistency, integrated graphics, accessible detail, and hierarchy of chunking and labeling)

  ◊ Presenting the information on paper and on-line, based on the type of information and its purpose for the user

2.3.5 INFRASTRUCTURE

The purpose of an infrastructure is to model the mixture of technical components (hardware, software, and communications facilities), their geographic distribution, and how they support or implement the business systems and other parts of the information system architecture environment within the Department. The infrastructure description conforms to the Technical reference model as described in the TISAF.

2.3.5.1 Key Aspects of Infrastructure Development

The infrastructure model

- Models the components and the interactions between them
- Provides a layered perspective

To develop the infrastructure, multiple components of the organization must be analyzed as depicted in figure 2-8.

Because the infrastructure provides the support and linkages for the other architectural components, it cannot be fully developed until the other architectural components have been defined and developed in some detail. It describes the available technology and the limitation of technology on work requirements, work patterns, and work methods. It focuses on the performance of the information systems in support of the organization and addresses standards and security issues.

To define the infrastructure, special consideration needs to be given to concepts being defined in the other architectural components, such as the work locations and user groups in the work architecture. However, by adhering to infrastructure principles and standards, one can scope the

infrastructure components using commercial off-the-shelf (COTS) software, compatible hardware and common hardware.

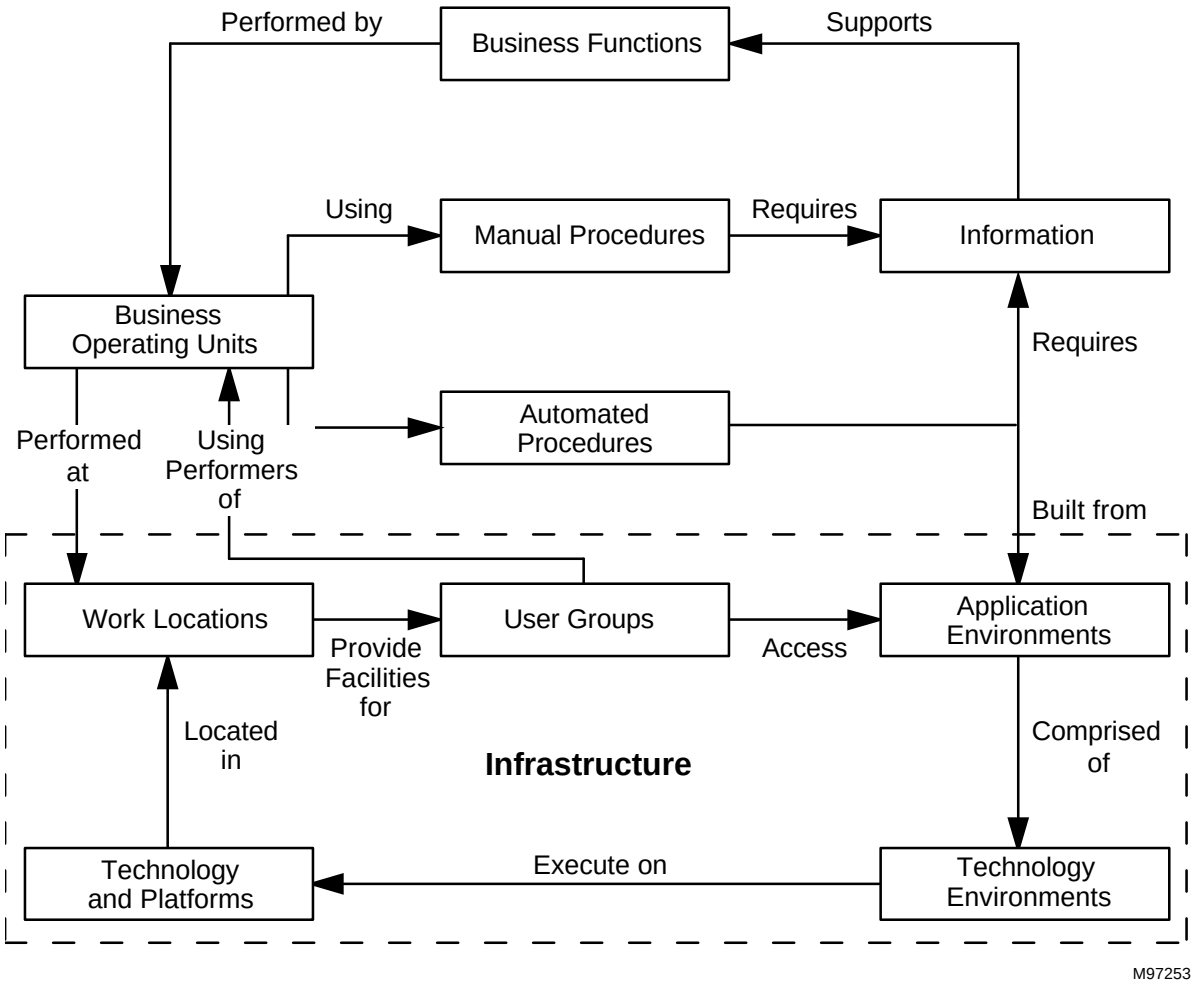

Figure 2-8. The Infrastructure Model

2.3.5.2 Possible Methods and Techniques for Infrastructure

The following approaches may be used to represent and model the infrastructure:

- *Layout diagrams.* Layout diagrams describe the various hardware and software components and the interacts between them. (See appendix C for more details.)

- *Layered diagrams.* Layered diagrams show the layers of hardware and/or software on a specific platform or component. (See appendix C for more details.)

- *Reference models.* Reference models define a standard organization for a set of integrated hardware or/and software components.

Table 2-3 presents a quick overview of the key concepts captured in each architectural view as well as examples the of some of the representations used to capture them.

Table 2-3. Architecture Views and Techniques

Architecture View	Work	Functional	Information	Infrastructure
Specifies	Work locations Organizational units User roles Communication coordination lines and dependencies between organizational units	Business processes (end-to-end) Business functional areas Functions performed	Data entities Data relationships Data types Data locations and volumes Data access and security	Platforms including • Software • Hardware • Telecommunication Standards Reference models
Example Techniques	Workflow models of key processes Business units Hierarchy chart of organization, Block diagrams BPR models	Object-oriented models Data flow and functional models IDEF0 Information engineering Scenario analysis State oriented models Simulation models	Data models Object-oriented models IDEF1x Information engineering Information mapping	Layout diagrams Layered diagrams Reference models

2.4 ARCHITECTURAL ISSUES

This section presents a set of issues important to a developer on a EISA effort. These issues include the architecture process, concerns and risks in defining an architecture, as well as the implementation of the architecture.

2.4.1 THE ARCHITECTURE DEVELOPMENT PROCESS

An information system architecture needs to be designed. A development process is generally used to sequence the activities as well as allow measurement of progress. It should embody a method which will ensure certain properties of the developed system.

Any particular design process must be tailored to fit into the particular set of organizational objectives. The choice of a design process is left up to the organization. The only prescription is that the organization use one or more development processes in developing its EISAs. Chapter 3 addresses the architecture development process in detail.

2.4.2 CONCERNS AND RISKS IN DEFINING AN ARCHITECTURE

The concerns and risks to address when defining an enterprise architecture include the following:

- First, not only the architectures but also the overall set of functional capabilities modeled may vary widely from architectural effort to architectural effort. A one-size-fits-all approach to an EISA will be difficult to create across these organizations.

- The existing legacy systems need to be carefully considered when modernizing an enterprise's information systems. First, in many domains legacy systems do not reflect a single evolutionary line of systems, but diverse systems, built for diverse principles. Architectures do not necessarily remain stable across these contexts (at least not in the absence of systematic reuse program that might provide the added incentive to hold to an established architectural framework).

  There is a risk that an EISA will be created based on architectures of legacy systems that may not scale up to be applicable across the entire range of systems desired for the business. That is, without some ability to evaluate the architecture relative to a set of business requirements there may belittle basis for deciding what features to preserve.

  The opposite risk results from not studying legacy systems adequately: in this scenario, systematic reuse amounts to little more than system design using a possibly wider repertoire of techniques for increased flexibility. That is, the architecture arrived at will be only as good as the architecture and design expertise of the domain engineers. Since domain engineering involves knowledge of many new areas of software engineering, the engineers will often not be the same people as those who know the workings of the existing systems and the rationale for their design.

- In addition to the risk of nonexperts developing these designs (possibly bringing their own representational biases, such as the assumption that an object-oriented or client/server based architecture is always preferable), the fact that the EISA will be applied across many systems means that the potential impact of a poor design decision will be much broader than if made within the context of a single system's development.

- The quality of an architecture is judged by how well it meets the mission of the business and how easy it is to tailor it to meet changes in the business objectives, business operations and the evolution of the underlying infrastructure components.

  The designer must achieve a proper balance between human, organizational, and technological issues. Metrics associated with quality parameters and the achievement of functional operations must be considered.

- The processes of verification and validation are important to the methodology of architectural development. These processes succeed only when the architectural development process is carried out within a framework that uses standardized procedures.

Validation bridges the objectives and concerns of the different parties involved in the ownership, definition, implementation, and implementation of an architectural solution.

- Reaching agreement in large, multicultural, competitive groups is difficult. The larger the group, the more difficult it will be to reach agreement. Thus, a durable commitment to adhere to an agreement once it is reached is necessary. Procedures for change to elements of the agreement in an orderly fashion must be instituted in order to maintain or modify assurances regarding the operation and nonoperational requirements of any system.

- Conflicts among the members of the group must be identified and dealt with. For example, there may be a conflict between requirements for the openness of the architecture and the confidentiality requirements arising from the security policy.

- Engineers commonly use the word "architecture" in two senses: the art of designing and building complex structures and the high-level design of the structures themselves. A good architecture defines the boundaries between the major components of a system. Furthermore, a good architecture should consist only of components that are either stable or controllable (Mowbray and Zahavi 1995, 55). As EISAs are developed, evaluate them with respect to the system qualities that they provide.

- The creation of an EISA is not a one-time activity nor should it result in a static and unchangeable representation. The EISA must be periodically revisited and updated so that it continues to be an effective business and information management resource.

An architecture, by providing explicit models, enables the identification and reasoning of enterprise level and system level concerns. These concerns are critical when attempting to develop an integrated set of systems that, as a whole, meet the business needs of an organization.

An EISA includes various architectures, such as work, functional, information and infrastructure which model the different perspectives of the overall architecture. The EISA is modeled a multiple levels of abstraction to provide the proper level of detail to capture the enterprise architectural vision. Finally, it is critical that formal modeling techniques are used to define and represent each of the architectures.

SECTION 3
ARCHITECTURE DEVELOPMENT PROCESS

This section describes a process for developing a complete, robust Enterprise Information System Architecture (EISA) which meets the needs of the Treasury Department and its bureaus. The goal of this process is to define an EISA by mapping business needs into information system descriptions that are bound by implementation constraints. By separating business needs from system requirements, it is possible to differentiate between functional requirements and implementation constraints. As a result, this process can be applied to developing an EISA as well as the individual ISAs that comprise the EISA. By defining the EISA before the system development life cycle (SDLC) begins, it is possible to define commonalities across functions, make principled choices among design alternatives, and determine architecture quality attributes such as performance and scalability.

This section does not provide a detailed methodology for describing all the attributes of an EISA nor does it provide a methodology for business process reengineering. It is not intended to provide the reader with a detailed understanding of the relevant methodologies and techniques, nor is it intended to be used to develop a single Treasury architecture for all bureaus. It is intended to be customized and adapted to meet each bureau's unique circumstances. It is intended to yield an architecture that can be used by each bureau to drive its own information system development activities.

3.1 ARCHITECTURE DEVELOPMENT METHODOLOGY

A *development methodology* is generally used to sequence the activities of the development process as well as allow measurement of progress. It should ensure that certain properties of the resulting system architecture are achieved. The development methodology must focus on the system as a whole. The system architect has the responsibility to know and concentrate on the critical details and interfaces that really matter to the success of the system and not become overloaded with low-level details.

No one single architecture development methodology is recommended for developing EISAs. The choice of a methodology is left up to the Departmental component. The methodology used should be appropriate to the level of architectural complexity required by each organization. The use of a formal structured analysis methodology is preferable to use of an informal analysis methodology.

Similarly, no suite of tools currently is recommended for describing and developing system architectures, but the use of suitable tools is strongly encouraged. In the future, the Treasury Architecture Working Group (TAWG) will suggest a set of tools for assisting in the architecture development process.

An appropriate test of the ultimate worth of an architecture development methodology is whether the resulting EISA enables the development of information systems for supporting the organization's business. An architectural development methodology also enables the organization to exchange information with other departmental components concerning models/architectural

views, data models with associated meta-data, network communications and protocol information, and architectural view descriptions that may enable increased component reuse, data-sharing, and applications interoperability (DOE 1997).

Treasury bureaus are strongly encouraged to develop information system architectures that improve the Department's ability to conduct business among departmental components and with its external customers and business partners.

The Treasury Architecture Development Process (ADP) is divided into five activities which are depicted in figure 3-1, each of which is further divided into subactivities:

Activity	Title
1	Develop/Update Baseline Architecture
2	Develop/Update Target Architecture
3	Develop Architecture Transition Plan
4	Develop Architecture Implementation Plan
5	Administer Architecture Development Process

These five activities are preceded by Activity 0—an initiating activity—that is required to begin the architecture development process. This activity assembles the project team, allocates resources, and develops the project plan. It is depicted in figure 3-1 as a separate activity because the actual ADP is an iterative process.

The ADP produces a baseline architecture description, a target architecture description, an architecture transition plan, and an architecture implementation plan. These are provided to the enterprise's information systems organization for further development and deployment through its system development life cycle methodology.

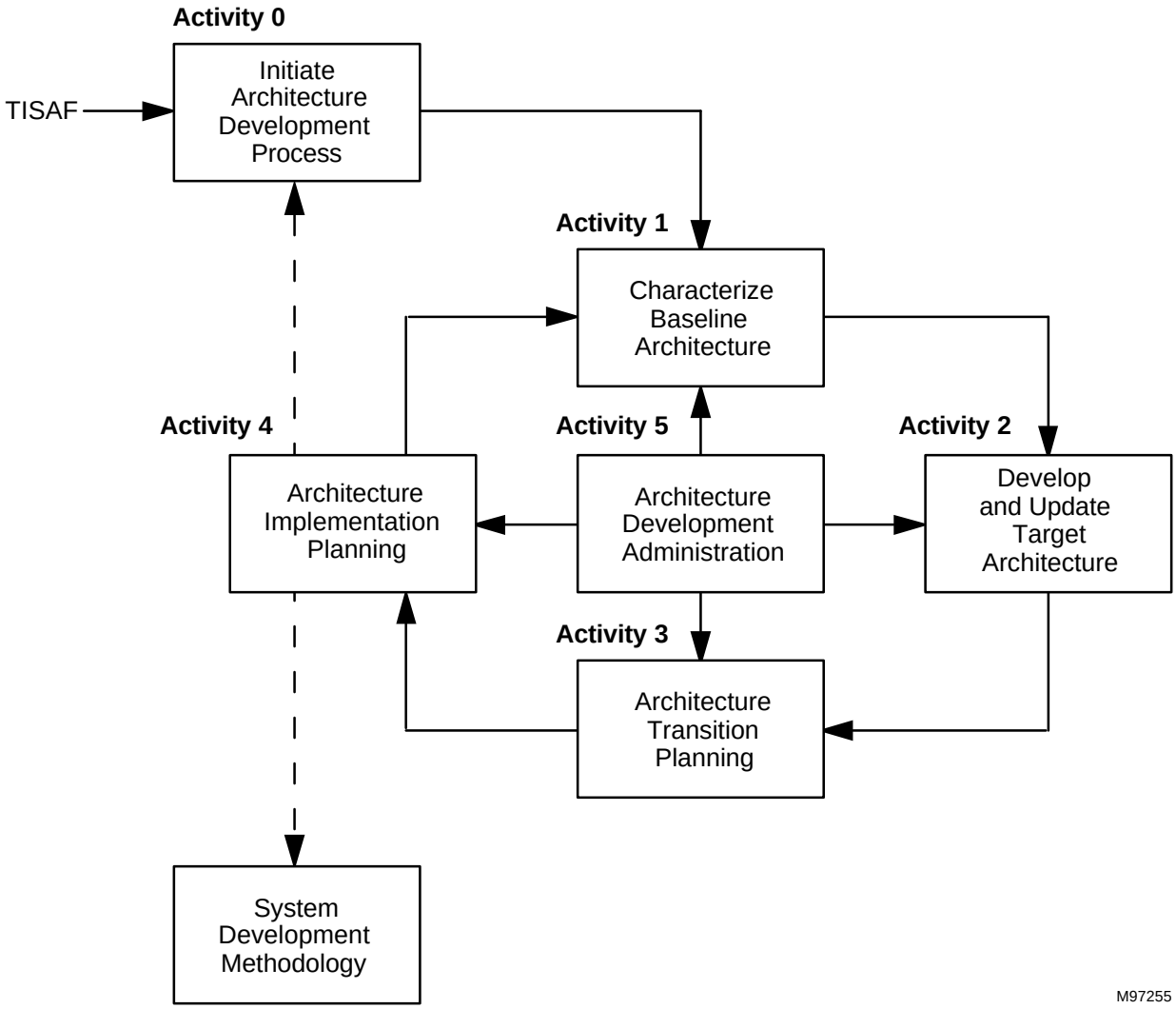

Figure 3-1. The Architecture Development Process

The ADP activities are described in more detail in sections 3.2 through 3.7.

3.2 ACTIVITY 0—INITIATE ARCHITECTURE DEVELOPMENT PROCESS

The initiating activity begins with the assembly of the architecture development team, the development of an architecture development project plan, and the allocation of the appropriate resources to support the effort. In addition, the architectural principles that drive the process must be customized from the set of architectural principles given in the TISAF.

The steps for initiating the architecture development process, depicted in figure 3-2, are described in the following subsections.

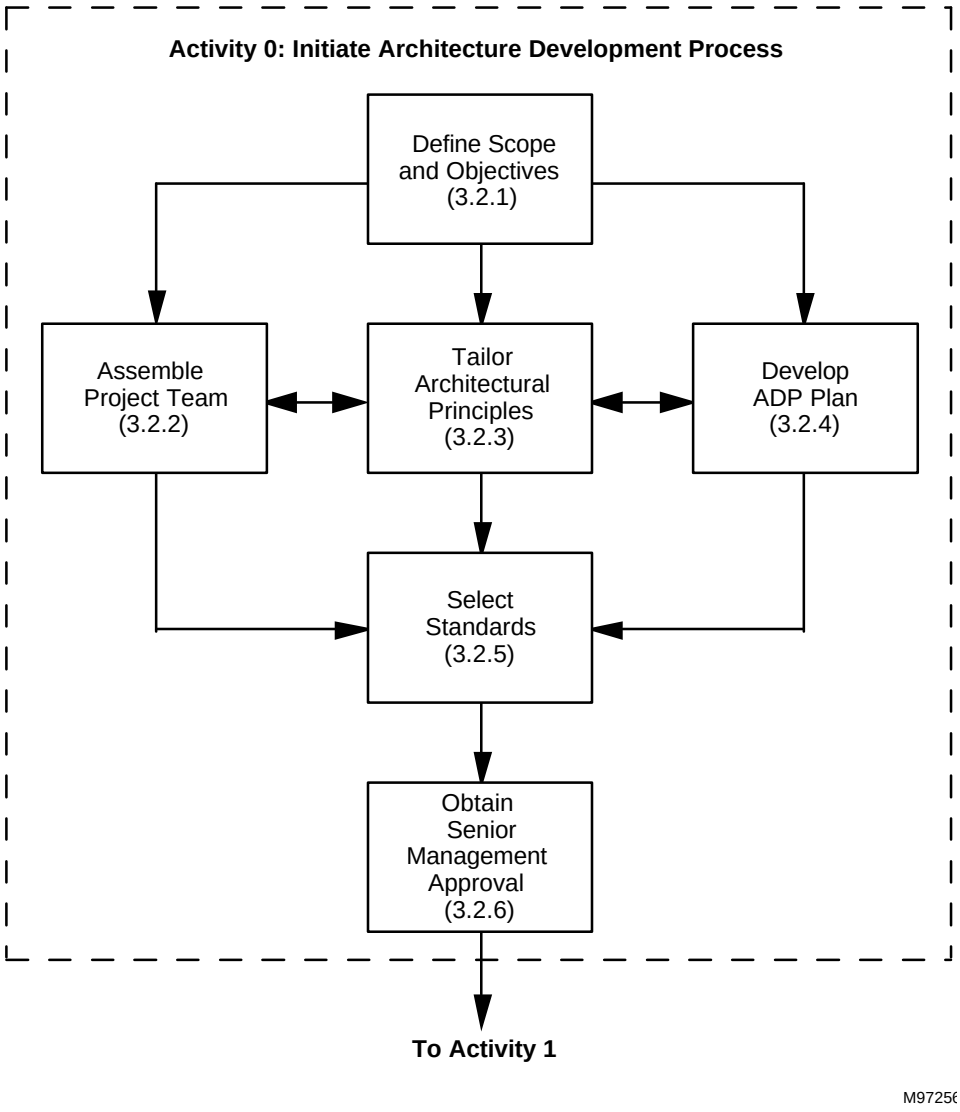

Figure 3-2. Initiating the Architecture Development Process

3.2.1 DEFINE THE SCOPE AND OBJECTIVES

An architecture development project should have a clearly defined charter. This narrative document establishes the project through the appropriate management approvals and specifies the scope, objectives, and expected outcome of the project. The scope should be defined in terms of organizational elements, mission areas, and/or functional areas that are to be addressed.

The charter represents an agreement between the senior executive sponsoring the project and the project leader. At the Departmental or bureau level, the senior executive is the SIRMO or Chief Information Officer.

As part of this activity, the following may need to be accomplished:

- Gather existing business/computing background material.
- Prepare a preliminary vision statement of the future enterprise/organization and the desired computing environment.
- Test/obtain reaction to the vision (statement/scope/objective) with internal/external information management specialists and senior management.

An example format for the charter is provided in appendix D.1.

3.2.2 ASSEMBLE PROJECT TEAM

An architecture development project must, of necessity, cross organizational and functional boundaries subject to resource constraints. Members of the project team should be drawn from all components of the organization.

The project leader is usually appointed by the senior executive sponsoring the project. The senior executive is also responsible for providing the resources (personnel, finances, equipment) as required for the success of the project. The senior executive will review the project's results and makes the go/no-go decision or a go/no-go recommendation to the rest of the organization's senior management based on the results of the project.

The project leader develops the project charter in conjunction with the senior executive. The project leader identifies the prospective team members on the basis of skill, business knowledge, or technical expertise. Once the architecture development project begins, the project leader manages the team on an ongoing basis. The project leader communicates the project status, problems, and accomplishments to the senior executive.

Two types of project team members may assist in architecture development: core team members and subject matter experts. Core team members should be dedicated to the project on a full-time basis. Most of the core team members should come from the mission and business areas, but several should come from the information system organization. Business area members will focus on the business needs of the enterprise, while the information systems personnel bring knowledge of the information and technology resources to the team.

The number of core team members will vary with the size of the organization, the scope of the project, and the anticipated breadth and depth of the results. An average size of six to eight core team members is suggested.

The core project team can be augmented by subject matter experts from individual business or mission areas on a part-time basis. Subject matter experts will bring specialized knowledge about business processes and requirements to the attention of the core project team.

Ideally, project team members should be selected to have the following attributes:

- Knowledge of the relevant mission-critical areas within the project scope
- Good oral and written communication and presentation skills
- Ability to think at several levels of abstraction and comprehend futuristic concepts

The core project team may require the assistance of support personnel, including clerical support, librarians, and technology experts. These individuals may be permanent support staff or acquired on as-needed basis.

A sample memorandum and roster of ADP team members is provided in appendix D.2.

3.2.3 TAILOR ARCHITECTURE PRINCIPLES

A set of architectural principles for guiding the development of EISAs within the Treasury and its bureaus has been provided in the TISAF, section 3. For departmental components, these architecture principles should be refined to meet the specific business needs of the component. Additional architectural principles may be added to the set as required by the individual departmental components.

These architectural principles become the overall architectural guidance that the ADP team will use in analyzing its baseline architecture and developing its target architecture in later activities of this process.

3.2.4 DEVELOP ARCHITECTURE DEVELOPMENT PROJECT PLAN

The project leader develops an ADP plan using the project charter for guidance. The plan defines the work breakdown structure (WBS) of the tasks to be performed and the resulting products and documentation that the project team will produce. Guidance regarding required products is provided in later sections of this chapter. It also identifies any assumptions about the nature of the effort and any limitations that may affect successful completion of the project.

The project plan should include a schedule of the tasks to be performed and include scheduled project reviews with the senior executive. It should include periodic internal team reviews and lead time for product and documentation preparation including a comment period for the reviewers of formal documentation.

The project plan should provide a rough estimate of the level of effort for core team members and subject matter experts, the time required, and the estimated cost. Potential risks should be clearly identified (e.g., tight schedule, unavailability of personnel, limited funding). The budget for the project should be negotiated with and approved by the senior executive early in the project.

Among the elements of the project plan to be determined are:

- Who the stakeholders are in the process and what their “win” conditions are; e.g., what set of functions and features will make each stakeholder feel that his needs have been met

- What problems need to be solved and/or fixed
- What priority should the different problems receive, if this is known *a priori*
- What documentation and publishing standards should be used
- What additional resources, personnel with appropriate skills and knowledge, and tools need to be brought into play
- How the available resources, people, and tools are allocated to the different activities of the ADP plan, including a work breakdown structure, if needed
- What the products (deliverables) are and the schedule for completion and delivery
- An initial list of source documents to be used in the analysis

Tools for preparing graphics, developing information repositories, and performing analysis should be selected. Access to individual workstations and other computer resources is necessary for each core team member; part-time project team members may share access to community workstations.

The ADP team should be trained in the use of all tools to ensure that each team member is comfortable working with the tools and can be productive without extensive clerical support. Not all team members require training in all tools, so the project leader must determine which tools a team member is most likely to use and arrange for training or refreshment in the use of those tools.

Documentation and publishing standards may have already been determined by the organization. These should be followed in preparing the architecture documentation. If not, these should be established by the project leader and documented in memoranda.

The project team needs dedicated individual office space, meeting rooms, and occasional access to larger conference rooms for special meetings and reviews. Other considerations include office space with erasable white boards, including one that can automatically make photocopies of its contents. Appropriate computer resources should be placed in each individual's office and the team meeting rooms with access to the selected tools. Individual offices require ample shelf and file space. In addition, one room should be designated as the project library and should contain at least one copy of every project source document.

The architecture should be developed and maintained in an electronic format which can be used by all elements of the organization. Succeeding versions of the architecture need only be updated from the original version.

A suggested outline for the ADP plan is presented in appendix D.3.

3.2.5 SELECT STANDARDS

Standards are the "glue" that enable users to interoperate seamlessly across applications, platforms, and organizations (DISA 1997). In the current technology environment users are confronted with islands of automation—myriad and redundant computer systems that have been used to automate nonstandard, and frequently inefficient, functional processes. Over time, users have realized that standards can promote the interoperability necessary to improve business operations. As the TAFIM puts it, the "open systems train has left the station and it will not turn back."

A standards-based architecture defines reusable and interchangeable architecture components that promote flexibility and modularity in the architecture. These components may be supplied by any number of vendors whose products conform to the standards. Selection of standards is a difficult process because some standards may conflict with one another. Standards that are selected for the target architecture may conflict with the features of legacy systems.

The project team should select an appropriate suite of standards from the standards profile specified in the TISAF to assist it in developing the EISA. It is not necessary to select all of the standards that will ever be used in implementing the target architecture at the beginning of the project.

3.2.6 OBTAIN SENIOR MANAGEMENT APPROVAL

Once the previous steps have been completed, it is essential to the success of the project to receive approval from senior management so that the products and documentation will be well-received. The senior executive is responsible for obtaining a commitment from other senior management personnel on the availability of needed resources and information.

An example memorandum for this purpose is provided in appendix D4.

3.2.7 SUMMARY OF ACTIVITY 0

The steps, inputs, and outputs of Activity 0 are summarized in the table 3-1.

3.3 ACTIVITY 1—CHARACTERIZE BASELINE ARCHITECTURE

A *baseline architecture* is the "as-is" description of the enterprise's existing architecture. This activity determines where an organization is situated architecturally. It is not an operational review or audit but more of an assessment and characterization of the current environment. It is used to establish a baseline or starting point for architecture development. The TISAF provides an effective means for organizing this review and presenting the current status.

Table 3-1. Activity 0 Summary

Activity	Inputs	Work Products	Deliverables

Define scope and objectives	Organization's mission statement TISAF	ADP team charter	None
Assemble project team		Memorandum assigning the ADP team members A roster of the ADP team members	None
Tailor architectural principles	TISAF	A tailored set of architectural principles	None
Develop ADP plan	Existing tools, if any	Assignments for ADP team Members Memoranda listing tools to be used during the project	ADP project plan
Select standards	Organization standards TISAF	Memoranda listing standards to be followed during the project	None
Obtain senior management approval	ADP team roster ADP project plan Budget requirements	Memorandum authorizing ADP	None

The baseline characterization phase results in a picture of the existing architecture along the four TISAF views: work, information, functional, and infrastructure. The term "characterization" is used because the data gathering and analysis are not exhaustive. It is not necessary, nor is it desirable, to expend the time and effort to document every detail of the current architecture. Only enough detail is gathered to allow informed decisions to be made with regard to the desired target architecture (DISA 1997).

The baseline architecture describes the current inventory of information systems and their components. This inventory is useful for

- Identifying hidden assets, gaps, and redundancies
- Managing business costs
- Finding out who is using what and why
- Classifying by value the business assets (related to information technology (IT))

The baseline architecture provides insight into the existing complement and configuration of information processing resources and information technology utilization. Without a baseline architecture, it will be difficult for an organization to develop effective implementation plans that can lead it to its desired target architecture. A clear view of the current architecture allows the organization to begin to identify opportunities for change as it refines its vision of the target architecture.

In summary, the purpose of the baseline architecture is to

1. Establish where the organization is in order to measure the difficulty of getting to where it is going

2. To measure progress in infrastructure development during the transition period

3. To identify redundancy among existing systems and their components

4. To identify incompatibilities among existing systems and their components

5. To identify common functions and information sources that may be coalesced

The major deliverable of this activity is the baseline architecture document. This document characterizes the existing environment in sufficient detail to enable the organization to begin the requirements analysis process for the target architecture. The amount of detail required in this document depends on the need of the organization; the degree to which it intends to modify, enhance, or replace existing information systems; and the overall schedule of the project.

The steps comprising the development of the baseline architecture are depicted in figure 3-3. Each activity is discussed as if it were being performed for the first time. However, differences between the initial definition and the successive updates are explicitly noted.

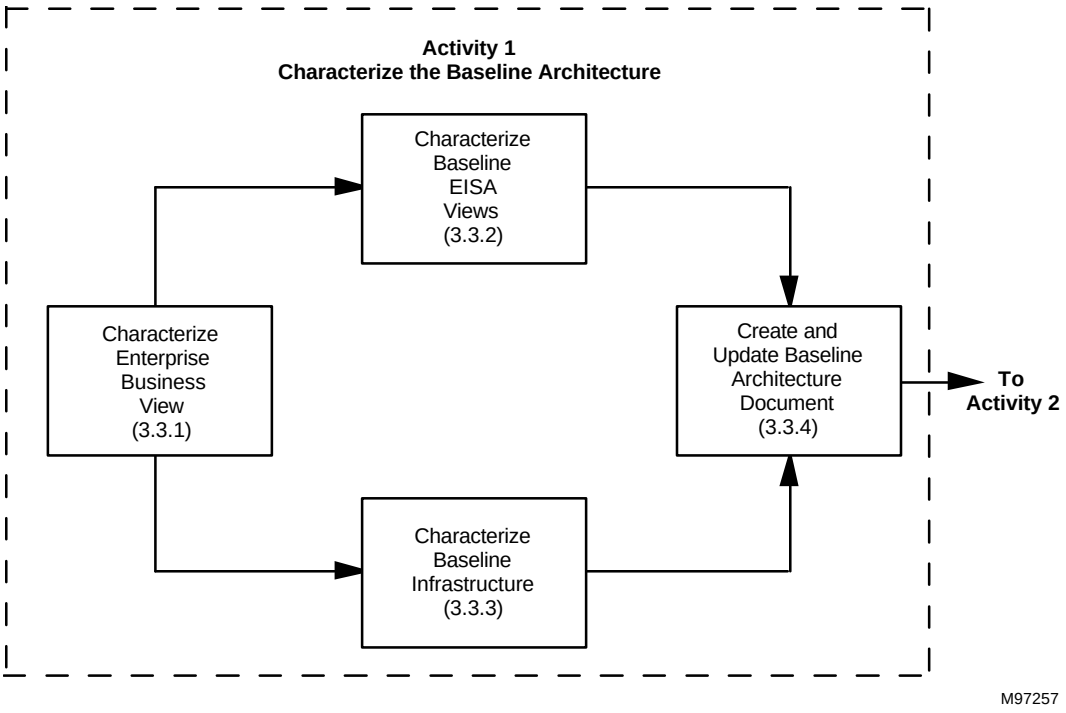

Figure 3-3. Characterizing the Baseline Architecture

3.3.1 CHARACTERIZE THE ENTERPRISE BUSINESS VIEW

This step either defines the enterprise's business view, if no definition currently exists, or updates the current view based on events occurring during the previous iteration of the architecture development process and corresponding system development process. In the previous period the organization may have reorganized, added new missions, revised business processes and functions, or relocated business operations. Any or all of these actions may cause revisions to the organization's business view.

Some of the information needed to define the "as-is" structure can be gathered from existing documents. For example, the formal organizational structure is normally documented in detail, and mission statements and strategic plans are also usually available. However, considerable effort may be required to gain consensus on some elements of the business view such as the major objectives of the organization, the critical success factors that measure accomplishment of the objectives, and the factors which inhibit accomplishment.

This step should be performed relatively quickly as its work products will be refined and enhanced during the development of the work architecture described in section 3.3.2.1.

3.3.1.1 Describe the Organizational Structure

The organizational structure reflects the mission and objectives of the enterprise. During this step, organizational charts are reviewed and updated. A primary focus of this step is to get a physical view of the organization in order to have an accurate view of the organization's business and structure.

An enterprise is usually organized into a set of business operations, some of which are externally directed to provide services and products to the enterprise's customers, and some of which are internally directed to provide infrastructure support to the other components of the enterprise.

This information may be collected in an automated data repository as well as be presented in diagrams and charts. For large or detailed organizations, a narrative report may also be prepared for reference within the project team. Matrixes that relate customers and providers to products and services can succinctly present complex organizational structures.

3.3.1.2 Describe the Organization's Mission Objectives

The organization's mission objectives flow down from the enterprise level to business operations. Using the mission statements and other available documentation as a starting point:

- Relate the objective statements to the nodes in the organization structure.
- Identify the products and services produced or offered by the organization.
- Identify the internal and external customers (or clients, or stakeholders) that are recipients of the products and services of the organization.
- Identify providers and their products/services that are critical to the business.
- Relate customers/providers to major organization units through the products and services that are provided.

3.3.1.3 Define/Update the Business View

The business view describes the enterprise's business operations in terms of services and products. The work products of this step include

- The list of customers/providers
- A list of products or services
- Matrixes that relate

    ◊ The customers and providers to products and services
    ◊ Organization elements to customers and providers
    ◊ Organization elements to products and services provided

Additionally, the ADP team should develop an enterprise-wide diagram of business operations with links to customers/providers of products and services. The diagram should be annotated to reflect work locations. The enterprise-level diagram should be decomposed to reflect the internal structure of individual business operations. The degree of decomposition depends on the size and complexity of the organization, the breadth of the mission objectives, and the perceived future needs of the ADP team.

The ADP team should also produce a glossary of the terms and acronyms used in the diagrams and other work products.

A sample enterprise business view outline is provided in appendix C.5.

3.3.2 CHARACTERIZE BASELINE EISA VIEWS

This step develops the work, functional, and information architecture views of the Baseline EISA. The infrastructure view is developed in section 3.3.3.

To *characterize* means to describe as succinctly as possible the current state of computer-based automated support for the enterprise's business operations. Detailed, definitive architectural descriptions for the baseline architecture are not needed. Enough information is required to determine what information systems and data an organization has in order to plan for what it needs.

3.3.2.1 Characterize the Baseline Work Architecture

In order to develop the information systems that can support the business operations of an organization, a work architecture is required that identifies and describes the major operations that are performed by work groups in support of functions. It defines the types of work (logical working units) in terms of the types of workers (classes of IT users) and types of work locations (places where the functions of the organization are carried out).

The work organization view should be independent of line organization design. Many traditional IT solutions were tailored to specific line organizations, resulting in hard boundaries and

inflexibility. Work organization modeling recognizes the realities of "networks" of individuals and their supporting automated and manual systems. It supports the team concept, the multiple roles (or team memberships) that individuals can have, and recognizes that teams can be composed of members who work remotely from each other.

It also should recognize external users and external functional locations. Key external constituencies (e.g., legislative organizations such as Congress) and suppliers are obvious candidates. Employees working from home office locations or while traveling should also be considered for inclusion.

The work organization view helps to describe the before and after impacts of technology on the organization. It becomes the basis for detailed redesign of work processes, communication programs, and user training to address change management requirements.

The above description is taken from the TAFIM, Volume 4 (DISA 1997).

3.3.2.1.1 Identify the Work Entities

Work entities are objects which represent sets of information that can be manipulated by the business functions or providers of information. Identifying the work entities is the process of defining the business. This step is internally focused on the enterprise's business operations.

The ADP team should develop diagrams and charts of the work entities that fall within the scope of the organizational unit. It is useful to start with an enterprise-wide diagram or chart that can be successively decomposed to show suborganization structure. The depth to which work entities are decomposed depends on the needs of the ADP team. Also, the amount of information collected for each work entity depends on the needs of the ADP team.

Once this effort is completed, the ADP team can map business functions to organizational units and map customers and providers to organizational units.

This information is often best represented as a set of matrixes.

3.3.2.1.2 Identify the External Work Entities

External work entities are producers or consumers of some of the business entities defined in the previous section. External work entities provide input, receive output, or otherwise interact with internal work entities. They operate outside the boundaries of the enterprise. This step and the previous one serve to identify the boundaries of the enterprise.

Once this effort is completed, the ADP team can

- Map incoming/outgoing products/services to organizational units
- Map external customer/provider to product/services
- Develop entity-relationship map (internal and external contacts/customers)

This information may be collected in an automated data repository as well as be presented in diagrams and charts. For large or detailed organizations, a narrative report may also be prepared for reference within the project team. Matrixes that relate customers/providers to products and services can succinctly present complex organizational structures.

3.3.2.1.3 Identify the Interactions between Internal and External Entities

Interactions are information flows that occur between internal and external functions and entities. These information flows can take many forms, including both manual and electronic forms. These interactions cross the boundary between the organization and its environment.

Once this step is completed, the ADP team should be able to trace the flow of information associated with specified business entities throughout the organization as well as to external entities.

3.3.2.1.4 Document/Revise the Baseline Work Architecture

This phase defines/updates the baseline work architecture description. It should include a description of the major entities comprising the enterprise's business operations.

A work architecture description is neither right nor wrong, but can be good or poor depending on how well it describes how the business operations are supported by the business applications. A good work architecture is understandable, complete, consistent, and stable.

A sample baseline work architecture is presented in appendix C.6.

3.3.2.1.5 Review the Baseline Work Architecture

After the work architecture has been described, it should be reviewed with senior management and selected users and data owners to ensure a consistent, complete capture of the enterprise's organization.

The work architecture description should be circulated to the reviewers prior to meeting to review its contents. During the meeting the major business operations can be briefed to the reviewers. Comments from the reviewers should be used to revise the work architecture description.

3.3.2.2 Characterize the Baseline Functional Architecture

The functional architecture shows which functions of the organization can be supported by IT applications. It provides a high-level description of these applications. It also shows logical dependencies and relationships among functional areas. This architecture defines the scope and interfaces of applications and provides the basis for enhanced or added functionality. It identifies specific work groups and users of applications, their relationships to information, and their placement or possible distribution across types of locations and technology platforms. (DISA 1997).

The purpose of this step is to analyze current business applications that support the current business functions and to document data and information entities used by the applications.

This effort results in a high-level description, not a specification. It should be correlated with the development of the information architecture. It may occur concurrently with the infrastructure survey of paragraph 3.3.3.

3.3.2.2.1 Provide Functional Modeling Training

The ADP team members should receive training or refreshment in the functional modeling methodology used to develop the functional view.

This step may be skipped if the team membership has not changed since the last iteration or if the duration of ADP iterations is relatively short.

3.3.2.2.2 Characterize Current Business Applications

This phase collects and organizes information about the current business applications that are used by the business operations. Information that may be collected about the existing business applications includes

- Application name and description
- Application location (which platforms it is executed on)
- Work locations supported by the application
- Data stores used by the application
- Personnel/organization responsible for maintaining the application
- Primary users of the application (by business operation)
- Business functions supported by the application
- Resources allocated to support and maintain this application

The application description should be a brief, English-language explanation of what the application does, not how it does it. As with other products, definitions are neither right nor wrong, but are good or poor depending on how adequately they describe the existing applications. Reviewing these definitions with senior management, developers and maintainers, and users will help to clarify the descriptions.

The identification of files and databases used by each application overlaps with the data collected for the baseline information architecture.

Each organization must decide to what depth it will survey its applications. Applications range from small, simple programs to large multimodule, distributed systems.

3.3.2.2.3 Characterize Current Application Relationships

Business applications are correlated with the business functions they support. A matrix depicting which applications support which work entities from the baseline work architecture can be

developed to assist in identifying redundancies. Business functions not supported by any current application should also be highlighted. This allows business applications to be related to organizational units.

3.3.2.2.4 Characterize Functional Areas

A *functional area* is defined as a cohesive grouping of business functions. A functional area provides a basis for standardizing the descriptions of specific services and functions. It enables effective communication between the developers and users of the system as well as other organizations that may have similar business needs and requirements. This standardization will also enable an effective mapping between the business needs and requirements and the resulting architecture. There may be multiple functional areas within a business operation.

Functional areas are identified and defined through functional partitioning. For each functional area, the design issues, tradeoffs, and design rationale are identified and documented. A functional partitioning specifies the distinct functions or services that a functional area must provide, but not the actual structure of the software that provides those functions. Different functional areas may combine different functions. Two or more functional areas may provide the same service. The functional partitioning acts as a means for identifying common functions among applications.

3.3.2.2.5 Characterize Functional Area Relationships

The relationships, interactions, and dependencies among the functional areas are specified. These relationships can include data flow, inheritance, aggregation, and messaging.

One method for performing this step is to construct usage scenarios that describe how certain business operations are performed. The functions and entities are correlated with the business operations through the scenarios. It is important to document and describe the activities of the organization's business (logical) and work units (physical).

A diagram showing which functions are performed by which business units should include a brief narrative and a glossary of the terms used. It should be decomposed to no lower than three levels because it is intended to be correlated with the infrastructure descriptions developed in section 3.3.3.

3.3.2.2.6 Document/Revise the Baseline Functional Architecture

This phase defines/updates the baseline functional architecture description. It should include a description of the major functions—both automated and manual—performed by the business operations and the business applications which support the automated functions.

A functional architecture description is neither right nor wrong, but can be good or poor depending on how well it describes how the business functions are supported by the business applications. A good functional architecture is understandable, complete and consistent, and stable.

An example baseline functional architecture is shown in appendix C.7.

3.3.2.2.7 Review the Baseline Functional Architecture

After the baseline functional architecture has been described, it should be reviewed with senior management and selected users and data owners to ensure a consistent, complete capture of the application resources and how they support the business functions. The baseline functional architecture description should be circulated to the reviewers before the meeting for review. During the meeting the reviewers can be briefed about major applications. Comments from the reviewers should be used revise the baseline functional architecture description and the application definitions.

3.3.2.3 Characterize the Baseline Information Architecture

The information architecture describes the information used by the organization and the relationships among collections of information (subject databases). It is important to include all forms of information and types of media in this view. Again, placement and distribution to working locations in support of user and application access is a key consideration. (DISA 1997)

The purpose of this step is to document the major kinds and interrelationships of data and information needed by the organization, and to capture the corporate data model and data dictionary.

A well-defined information architecture facilitates data exchange and access, provides redundancy control, reduces the impact of changes, minimizes data handling, improves data integrity, and facilitates systems development productivity. Formation of an information architecture can be divided into two areas: (1) information management and (2) data components.

- Information management encompasses not just raw data but objects not traditionally thought to be data such as voice and images (including documents and moving images).

- Data components include items which are more technical such as the data model, data dictionary, file structures, programming languages, storage (capacities, access methods, media, techniques), dissemination/distribution, backup and recovery, integrity, and security. From a corporate point of view, the most important of these are the data model and the data dictionary.

Information management ensures that information is well-organized, centrally documented, secure, and available to those who need it. Some of the issues in information management are records management, data centers (any type of repository), data and information flow, ownership, migration, exchange/interchange, manipulation, duplication, replication, synchronization, standardization, contingency/disaster recovery/continuity of operations, accessibility, vulnerability, portability, etc. Other issues include linking business practices to a strategic vision and to customer needs, providing data to the right people especially critical data, overcoming political agendas and turf issues, obtaining commitments, and ensuring accurate and complete data.

Additionally, one major issue in developing an information architecture is data dependency; that is, the types of data needed for mission accomplishment. Data is not information until it is made

meaningful. As a corollary, data repositories (which include information systems, as they may contain data) are not useful until the data they contain is put in a meaningful form. When determining a definition for *meaningful information*, the primary component is mission—does the information support the mission of the Department, organization, program, etc.? As an addendum, regardless of the specific purpose of the information, all information should ultimately support or contribute to the support of the Departmental business areas discussed in the Department's strategic plan.

Data can reside in a multitude of places and platforms. However, the key in achieving the business objectives is determining what data is to be stored and how it is to be organized. This is the purpose of data modeling.

3.3.2.3.1 Provide Data Modeling Training

The ADP team members should receive training or refreshment in the data modeling methodology used to develop the information view.

This step may be skipped if the team membership has not changed since the last iteration or if the duration of ADP iterations is relatively short.

3.3.2.3.2 Develop/Update Corporate Data Model and Data Dictionary

The ADP team develops/updates the corporate data model and data dictionary (CDM/DD) which reflect enterprise-wide data and information resources.

The CDM/DD is a list of current data entities (people, places, things, events)used by the enterprise. The breadth and depth of this list (e.g., the amount of detail provided) depends on the number of data and information entities, the size and complexity of the enterprise, and the degree of detail required to begin the next iteration of the ADP. Within the CDM/DD, data and information entities are associated with business functions, business customers/providers (from the business view), and organizational work units (from the Work Architecture).

The purpose of the CDM/DD is to create standard definitions and descriptions for each data and information entity in the information architecture. Graphical descriptions of the entities' relationships are drawn. Detailed matrixes are prepared to relate these entities to business functions. These matrixes can be used to validate and verify the matrixes previously prepared for the business view in section 3.3.1.3 and section 3.3.2.2.3. External data should be excluded from these matrixes unless the data is used internally within the enterprise.

The ADP team should review the CDM/DD descriptions and matrixes to determine whether redundant information is being retained within the organization. The ADP team should also review entities to determine whether their definitions are consistent with other entities, particularly where one entity is transformed into one or more other entities. This information is not acted on immediately, but is presented to the senior management and the reviewers of the information architecture (see section 3.3.2.3.4) for decision.

As Spewak notes, the purpose of this step is not to define databases, but to understand the primary data and information entities that an enterprise uses in conducting its business operations (Spewak 1993). Therefore, the descriptions comprising the information architecture cannot be deemed right or wrong. However, an information architecture may be of good or poor quality. A good information architecture is understandable, complete, consistent, and stable.

3.3.2.3.3 Document/Revise the Baseline Information Architecture

After the CDM/DD has been created along with the CRUD matrixes and the entity relationship diagrams, the baseline information architecture description should be prepared which presents the data and information entity definitions, their usage, and their relationships. Additional information that may be provided includes sizing and volume information, backup and recovery requirements, and integrity and security constraints.

A sample baseline information architecture is provided in appendix C.8.

3.3.2.3.4 Review the Baseline Information Architecture

After the information architecture has been described, it should be reviewed with senior management and selected users and data owners to ensure a consistent, complete capture of the data and information resources.

The information architecture description should be circulated to the reviewers before the meeting for review. During the meeting the CDM/DD can be briefed to the reviewers. Comments from the reviewers should be used to revise the information architecture description and the CDM/DD.

3.3.3 CHARACTERIZE THE BASELINE INFRASTRUCTURE

The objectives of the current infrastructure analysis include the following:

- Identify the formal data stores and computer systems available to the organization.
- Identify the current level of computer support of the applications and the potential for future automation of processes.
- Assess the strengths and weaknesses of current computer systems.

This step surveys the information systems that are currently used by the enterprise. Alternatively, if such a survey already exists, it may be updated and revised using information from the last iteration of the architecture development process.

Because information systems are based on hardware, software, and telecommunications components, this survey gathers the data that can be used to describe the infrastructure view of the EISA.

3.3.3.1 Conduct Current Infrastructure Survey

This phase surveys the existing infrastructure within the organization in order to create an infrastructure inventory. The survey may focus on the following areas (DOE 1997), among others:

- Platforms
    - ◊ Intelligent wide area network (WAN) systems
    - ◊ Establishment-based switching systems
    - ◊ Local area network systems
    - ◊ Enterprise or corporate processing systems
    - ◊ Divisional or departmental processing systems
    - ◊ Desktop or portable intelligent workstations
- Applications environment
    - ◊ Batch processing
    - ◊ Computer conferencing
    - ◊ Decision support
    - ◊ Document processing
    - ◊ Document storage and retrieval
    - ◊ Electronic mail
    - ◊ Electronic publishing
    - ◊ Enhanced telephony
    - ◊ Expert systems
    - ◊ Hypermedia processing
    - ◊ Inquiry processing
    - ◊ Real-time control
    - ◊ Shared screen teleconferencing
    - ◊ Text processing
    - ◊ Transaction processing
    - ◊ Video processing
    - ◊ Video teleconferencing
    - ◊ Voice mail
- Technology environment
    - ◊ Communications management
    - ◊ Conferencing management
    - ◊ Database management
    - ◊ Development management
    - ◊ Distribution management
    - ◊ Document management

- ◊ Hypermedia
- ◊ Repository
- ◊ System management
- ◊ Transaction management
- ◊ User interface services

For each functional area identify and specify the computer systems that execute the applications of that particular work area. One or more computer systems may support a functional area.

Additional information that may be provided for each computer system includes

- The utilization of resources
- The physical aspects of the hardware, software, and telecommunications systems
- The programming languages, tools, and standards that have been used to implement the components of the computer system
- The rationale, if known, for each component of the computer system
- A catalog of existing information products, applications, software, databases, and technology infrastructure components
- A definitive list of planned and in-progress technology improvement projects
- An assessment of the current use of information technology to support the business activities

3.3.3.2 Correlate Infrastructure with Business Operations

This phase correlates the infrastructure's computer systems and telecommunications facilities with the business operations that own and/or use them. Each computer system is described in detail sufficient to identify the major hardware, software, and telecommunication components; the major applications executed on the systems; and the physical and operational locations of the systems. Matrixes are prepared showing the relationship of computer systems to business operations, computer systems to applications, and computer systems to physical locations.

3.3.3.3 Create/Update the Infrastructure Inventory

This phase develops the infrastructure inventory—a compendium of computer systems, their major components, the applications that execute on them, and the files and databases that reside on them.

It is recommended that the infrastructure inventory be maintained as a database for large organizations because of the expected number of computer systems and their components that are likely to be owned and used by the organization.

3.3.3.4 Create/Update the Baseline Infrastructure

The baseline infrastructure describes the computer systems at the hardware and software platform level. It gathers information from the previous steps in a narrative or other descriptive form of the computer systems used in the organization.

The narrative portion of the baseline architecture should be described in the services terminology of the TISAF, section 4. The reason for this approach is to make the description compatible with the target architecture, which will be described using this terminology and the TRM, and to map existing services to the corresponding standards of the target standards profile (TISAF, section 5).

A matrix mapping hardware and software platforms versus business applications should be created to show which business applications use which technology platforms. This matrix will show the extent to which different types of technology are used throughout the organization.

A sample baseline infrastructure description is depicted in appendix C.9.

3.3.3.5 Review the Baseline Infrastructure

After the baseline infrastructure has been described, it should be reviewed with senior management and selected users and data owners to ensure a consistent, complete capture of the hardware, software, and telecommunications resources.

The baseline infrastructure should be circulated to the reviewers before the meeting for review. During the meeting the infrastructure inventory can be briefed to the reviewers. Comments from the reviewers should be used to revise the baseline infrastructure and the infrastructure inventory.

3.3.4 CREATE/UPDATE THE BASELINE ARCHITECTURE DOCUMENT

The baseline architecture is the primary document describing the baseline architecture. The baseline architecture is a compendium of the work products of the previous steps of this activity. One outcome of describing and/or updating the baseline architecture is an understanding of how well the current suite of information systems supports the business operations of the organization.

To fully assess the adequacy of and satisfaction with current systems, user surveys or interviews should be prepared and conducted. From the surveys and interviews, a systems satisfaction chart and a problem area report, which would outline the strengths and weaknesses of the systems and data stores, should be prepared. Some organizations may periodically review systems for adequacy and satisfaction and have reports on file.

A template for this document is presented in appendix C.10.

3.3.5 SUMMARY OF ACTIVITY 1

The steps, inputs, and outputs of Activity 1 are summarized in table 3-2.

3.4 ACTIVITY 2—DEVELOP/UPDATE TARGET ARCHITECTURE

A *target architecture* is defined as the desired "to-be" architecture, which will meet the business needs, at a specific point in the future as determined by the needs of the organization. However, the rapid pace of information technology improvements limits the extent of the infrastructure view of the architecture.

A target architecture almost always represents enhancements to an existing baseline architecture to provide improved support for existing business operations as well as adding new functionality to support new business operations.

Table 3-2. Activity 1 Summary

Activity	Inputs	Work Products	Deliverables
Characterize the enterprise business view	TISAF Internal organization documents	Updated organization charts Enterprise business view	None
Characterize baseline EISA views	TISAF Existing system documentation Enterprise business view	Customers and providers lists Products and services lists Correlation matrixes Glossary Baseline work architecture description Baseline functional architecture description Baseline information architecture description	None
Characterize baseline infrastructure	Existing system documentation	Individual computer system descriptions Various matrixes Infrastructure description	None
Create/update baseline architecture	Working documents for each view	None	Baseline architecture

The overriding principle of the target architecture should be system simplicity, which is achieved by

- Following organizational standards, using the TISAF standards profile
- Using off-the-shelf components wherever feasible and possible
- Keeping the architectural diagram simple (at most two pages)
- Ensuring correlation with the user/customer organizational structure
- Supporting extensibility

- Supporting maintainability
- Providing clear definition of high-value customer features

Some or all of these may not be achievable during the first iteration from the baseline architecture. However, at each successive iteration of the architecture development process, the organization should strive to achieve more of these features in the resulting target architecture.

The major deliverable of this activity is a target architecture that describes the future view of the EISA supporting the Department or organization.

The steps comprising the development of the target architecture are depicted in figure 3-4. Each activity is discussed as if it were being performed for the first time. However, differences between the initial definition and the successive updates are explicitly noted.

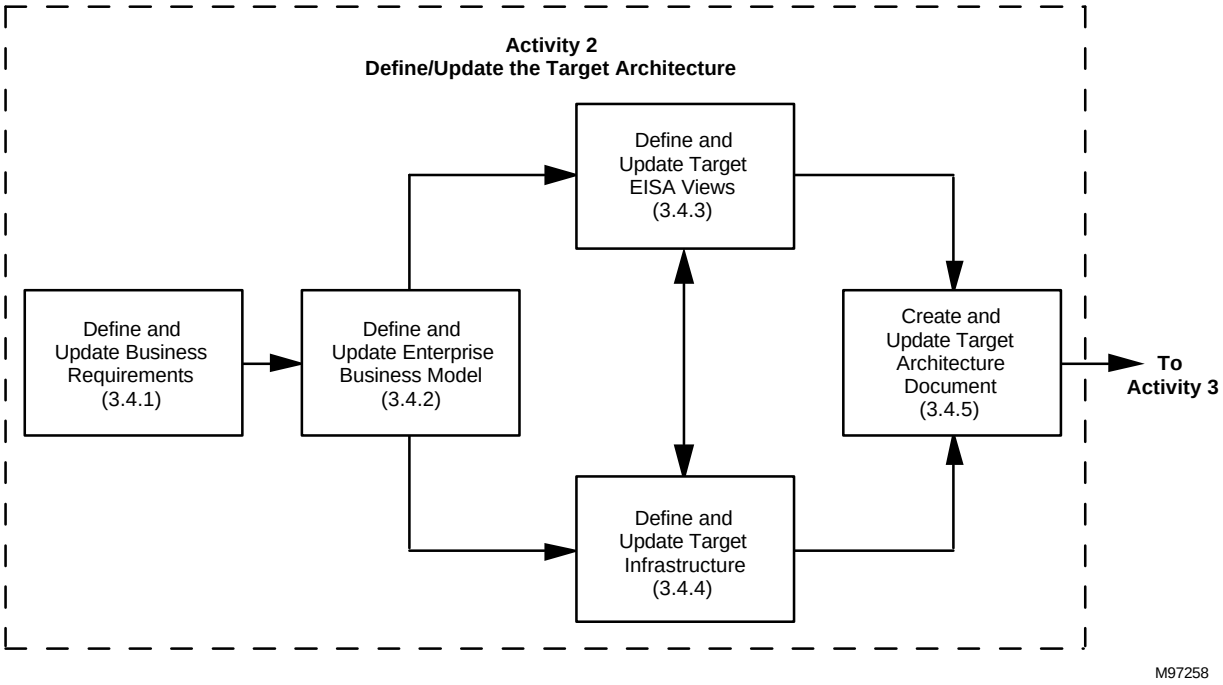

Figure 3-4. Defining the Target Architecture

3.4.1 DEFINE/UPDATE BUSINESS REQUIREMENTS

The target architecture in most cases builds on or evolves from the baseline architecture. It is driven by evolving business needs as well as by implementation, resource, technology, and schedule constraints. Some basic questions that should be answered while identifying the business requirements include the following:

- What are the strategic business objectives of the organization?

- What data/information is needed to support the business?

- Are the applications to provide that data/information in place?

- Is the technology to support the applications and the other subarchitectures for meeting the business needs in place?

- What does the organization need to do to evolve its information resources to the level required to satisfy current and future business requirements?

It is difficult to describe all of the business needs and operations of an organization in a single activity. Thus, iteration of steps 3.4.1.1 through 3.4.1.3 may be required.

Architecture development is not business process reengineering. Processes are only defined to identify business activities that may affect the EISA. The EISA may be used in the business process reengineering.

After this activity is concluded, the following products should be available:

- A definition of the scope of the architecture; put another way, a definition of the boundary between system and environment of each information system represented by the architecture

- The information that flows between each information system and its environment and the semantics of that information

- The set of features offered by each information system, e.g., the business needs it will support

- The behavioral and qualitative requirements to be met by each information system,

- Traceability from legacy system business requirements to new or updated target architecture business requirements

As part of the process of identifying target architecture requirements, it is necessary to identify and document the functional business requirements of the baseline architecture that are to be carried forward.

3.4.1.1 Collect Business Requirements

To answer the business requirements questions, the ADP team must analyze the mission(s) of the organization, the philosophy of the strategic decision-makers, the needs of customers and stakeholders, and the public image of the organization. The ADP team defines the strategic goals, problems, and critical success factors of the organization and develops the future picture of the business architecture. The result of this analysis is a set of business requirements that characterize how the organization wants to operate in the future.

Business requirements can be classified as:

- *Functional requirements.* Functional requirements are requirements that can be stated using a simple input-output model. An input changes the system state which results in an output that is generally independent of time. Intuitively, these requirements might be verified by watching the system perform in all of its states and comparing its outputs against some standard of correctness. The requirements might include safety, security, fault-tolerance, reliability, and availability, which are called functional quality attributes, because they enable the system to produce correct output in each state of the system's operation.

- *Performance requirements.* Performance requirements are time-dependent requirements that can be checked by observing the behavior of the system. In these requirements, changes to the system's state are dependent on time. Nonfunctional requirements lead to certain system qualities such as maintainability, openness, portability, etc.

The process of business requirements analysis is concerned with characterizing the desired or "to-be" information systems. For a specific system, these requirements may specialize the reference requirements and augment them with additional objectives, constraints, and evaluation criteria. However, the "to-be" system requirements also include the business requirements for retained functions and entities. These retained requirements should be clearly identified and documented. Also document business requirements that are no longer to be satisfied by the "to-be" system.

A crucial part of the design process is the exploration of the customer's need both to identify the stated needs and to uncover their implications (the unstated needs). By analyzing and refining the business requirements, the designers and the customer better understand the information system's requirements.

The output of this step is a business requirements document. This document should identify only the new or modified business requirements that have emerged since the last iteration of the ADP.

An example of a business requirements document is provided in appendix C.11.

3.4.1.2 Analyze and Assess Internal Functions/Entities

This step specifies the functions and entities that need to be changed to meet the business needs as expressed by the new or revised business requirements. The business requirements are correlated with the entities that represent the work architecture.

This process helps to identify the information system's drivers—those functions that must be implemented for an information system to operate correctly and successfully. Where data is stored is not that important. However, data stores are important for definition of technological needs for the applications and information sharing and distribution, especially for determining whether information can be accessed and used.

An assessment of current operations is a critical factor in determining future priorities and applications development projects. Existing business applications are assessed for supporting

future functions and information requirements. An assessment of the baseline architecture can be performed to determine (1) whether it is meeting senior management's objectives, and (2) whether senior management's assessment of information requirements can and will meet the business objectives. A review of the data and technology that supports the applications should also be made.

Some of the products that might result from this stage are:

- To-be-revised and planned data store descriptions
- To-be-revised and planned business process descriptions
- Data flow diagrams (not for building systems, but to understand business activities and interrelationships)

A data flow diagram that documents the data and information flow between business systems, data stores, and organizational work units—either internal or external—can be used to examine shareability and accessibility of information.

3.4.1.3 Define Issues, Risks, and Alternatives

One key to the successful development of the EISA is a systematic means of identifying risk elements and assuring that they are controlled. One of the most difficult steps in traditional systems development is to identify those areas that have the highest risk. Since most new information systems are built from the ground up, each component adds risk to the system.

Most design tradeoffs pit performance requirements against some other attribute. When performance is a driving requirement, it precipitates many major architectural design decisions (Clements 1995, section 2.1).

Functional requirements, on the other hand, do not usually yield driving requirements, because this would imply that attaining correctness of operation would be difficult.

This step documents architectural design issues, assesses their affect on the overall EISA, and develops a rationale for the selected alternatives. The result of this step is an architectural issues description that attempts to capture the issues arising from the business requirements and operations analysis. Among the questions that should be addressed in this document are:

- What interface and interoperability issues were raised during the business requirements analysis?
- What design issues arose from consideration of the constraints upon individual information systems?

- What alternative approaches were considered where significant issues were raised?

- What decisions were made where alternatives were available and why were they made?

3.4.2 DEFINE/UPDATE TARGET ENTERPRISE BUSINESS VIEW

The purpose of this step is to define the major future business functions and processes by identifying the functions and processes that employees do, or perhaps should be doing, irrespective of the organizational structure. Additionally, changes to current business functions must also be noted and recorded as they will affect the description of the target architecture.

The phases in this step are similar to those described in section 3.3.1 for defining the current business view.

3.4.2.1 Describe the Organizational Structure

For the target enterprise business view, the organization chart is reviewed for proposed or actual changes to business operations since the current business view was defined. The organization chart should be annotated to depict new, modified, or eliminated business operations. It is important not to remove eliminated business operations from the chart until the business view and architectural views are completely defined and/or updated.

The depth to which the enterprise is described depends on the needs of the architecture development project team. Among the activities to be performed are the following:

- Define the enterprise's organizational units.

    ◊ Review the organization chart.
    ◊ Identify the staff and their positions/titles/locations on the chart.
    ◊ Annotate other pertinent information on the chart.

- Define business locations (physical locations), if not within the same building.
- Relate business locations to the organizational units (who reports to whom).
- Identify ad hoc teams, working groups, committees, virtual organizations, etc.
- Identify organizational scope, responsibilities, objectives, goals, plans, etc.
- Match objectives to organizational unit.

3.4.2.2 Describe the Organization's Mission Objectives

The organization's mission objectives are reviewed for proposed changes, which may include new business areas as well as eliminating old ones (for example, programs have been discontinued). Where changes are detected, these changes are mapped to business operations. As a result of this mapping, new business operations may be required and old ones may be eliminated or modified in their own organization. Changes may also determine new services and/or products for existing or new business operations as well as the elimination of products and services from existing business

operations. Also, the customers and suppliers of individual business operations may change if new services and/or products are added or deleted from the business operation.

3.4.2.3 Document/Revise the Target Enterprise Business View

The list of work products related to the target enterprise business view should be reviewed for proposed changes to

- The list of customers and providers
- The list of products or services
- Matrixes that relate

    ◊ The customers and providers to products and services
    ◊ Organization elements to customers and providers
    ◊ Organization elements to products and services provided

The target enterprise business view has the same format as the baseline enterprise business view that is presented in appendix C.5.

Additionally, the ADP team should update the enterprise-wide diagram of business operations with links to customers/providers of products and services. The diagram should be annotated to reflect added or deleted work locations. The enterprise-level diagram should be decomposed into diagrams that reflect the internal structure of individual business operations. The degree of decomposition depends on the size and complexity of the organization, the breadth of the mission objectives, and the perceived future needs of the ADP team.

The ADP team should also update the glossary of the terms and acronyms used in the diagrams and other work products.

A lot of information about business operations is collected and analyzed during this step of the activity. This information should be documented in working papers and retained for future reference in the architecture development library (see section 3.7.4).

3.4.3 DEFINE/UPDATE TARGET EISA VIEWS

The target EISA views will be described through the architecture development process for the target architecture in the same manner as was done for the baseline architecture.

3.4.3.1 Define/Update Target Work Architecture

In order to develop the information systems that can support the business operations of an organization, a work architectures is required that identifies and defines the automated functions for the future enterprise.

3.4.3.1.1 Identify New Work Entities

The ADP team should develop diagrams and charts of new or modified work entities that fall within the scope of the organizational unit. An enterprise-wide diagram or chart can be successively decomposed to show suborganization structure. The depth to which new or modified work entities are decomposed depends on the needs of the ADP team. Also, the amount of information collected for each new or modified work entity depends on the needs of the ADP team.

The information to be collected may include

- The name of the work entity, a brief description of what it is, and what it does (e.g., business functions)
- The locations where the business function is performed
- A list of the subfunctions, if appropriate, of business functions performed by work entities
- The frequency of the business function, its inputs and output in terms of work entities, and the duration required to perform it
- Business entities, a description of their contents, where they are generally stored
- The organizational component responsible for creating, updating, or deleting the business entity
- A list of the business functions that use the business entity, and the actions performed
- The medium by which the information flows among the organizational components

In this step, both manual and automated business functions must be considered and work entities that reside on other than electronic media must also be considered.

Matrixes similar to those produced in section 3.3.2.1 should be produced in this phase.

3.4.3.1.2 Identify New External Functions/Entities

This step and the previous one serve to identify the revised boundaries of the enterprise as evidenced by changes to the business operations.

The information collected about external work entities may include

- The name of the external work entity, a brief description of what it is, and what it does (with specific reference to the enterprise)
- The locations where the business function is performed

- The frequency of the business function, its inputs and output in terms of business entities, and the duration required to perform it

- External business entities and a description of their contents, if individual components are used by several internal business functions

- The medium by which the information flows among the organizational components

Once this effort is completed, the ADP team can

- Map incoming and outgoing products and services to new or modified work entities
- Map new external customers and providers to product and services
- Develop an entity-relationship map (new internal and external contacts and customers)

This information may be collected in an automated data repository as well as be presented in diagrams and charts. For large or detailed organizations, a narrative report may also be prepared for reference within the project team. Matrixes that relate customers and providers to products and services can succinctly present complex organizational structures.

3.4.3.1.3 Identify the Interactions Between Internal and External Entities

This step identifies new interactions that will occur between internal and external entities. Among the activities to be performed are the following:

- Identify the goods, services, and information that flow across the boundary.
- Determine the direction of flow and the entities that participate in each interaction.

Once this step is completed, the ADP team should be able to trace the flow of information associated with specified business entities throughout the organization as well as to external entities.

3.4.3.1.4 Define/Update Work Locations

The ADP identifies all of the work locations of the organization. These are correlated with the business entities. Additional information describing the characteristics of the work locations, any unique features, and any specific constraints may also be gathered for inclusion in the work architecture.

3.4.3.1.5 Document/Revise Target Work Architecture

This phase defines/updates the target work architecture. It should include a description of the major locations where the business operations are performed and the organizational units which perform them.

The structure of the target work architecture is the same as that of the baseline work architecture which is described in appendix C.6.

3.4.3.1.6 Review Target Work Architecture

After the target work architecture has been described, it should be reviewed with senior management and selected users and data owners to ensure a consistent, complete capture of the work locations and processes. The target work architecture description should be circulated to the reviewers before the meeting for review. During the meeting major work entities can be briefed to the reviewers. Comments from the reviewers should be used revise the target work architecture.

3.4.3.2 Define/Update Target Functional Architecture

The purpose of this step is to analyze current business applications that support the current business functions, and document new or to be modified data and information entities used by the proposed applications.

This effort is a high-level description, not a specification. It should be correlated with the development of the information architecture. It may occur concurrently with the infrastructure survey of section 3.3.4.

3.4.3.2.1 Define Future Business Functions

This phase collects and organizations information about the future business functions that that will be used by the business operations. Information that may be collected about the future business functions includes

- Function name and description
- Function location (which platforms it is executed on)
- Data stores used by the function
- Primary users of the function (by business operation)

The function description should be a brief, English-language explanation of what the function does, not how it does it. As with other products, definitions are neither right nor wrong, but are good or poor depending on how adequately they describe the existing applications. Reviewing these definitions with senior management, developers and maintainers, and users will help to clarify the descriptions.

The identification of data stores used by each function overlaps with the data collected the target information architecture.

Each organization must decide to what depth it will determine whether future business functions. Functions range from small, simple programs to large multimodule, distributed systems.

Among the activities to be performed are the following:

- Define preliminary business functions.
- Decompose functions into subfunctions, if possible, but to no more than three levels.
- Group functions and subfunctions into a hierarchical business diagram or chart (i.e., diagram of major functions and subfunctions).
- Define the business processes and relate them to each other and to the functions, if applicable.
- Map business and information requirements and objects to major functions, processes, activities.

3.4.3.2.2 Define/Update Future Function Relationships

A matrix depicting which proposed functions will support which business functional areas can be developed to assist in identifying redundancies. Business functions not supported by any current application should be highlighted. This allows business applications to be related to organizational units.

3.4.3.2.3 Define/Update Functional Areas

New functional areas are identified and defined through functional partitioning. For each new functional area, the design issues, tradeoffs, and design rationale are identified and documented. The functional partitioning specifies the proposed functions that a functional area must provide, but not the actual structure of the software that provides those functions. Different functional areas may combine different functions. Two or more functional areas may provide the same service. The functional partitioning acts as a means for identifying common functions among applications.

Among the activities that may be performed are the following:

- Identify the new products and services produced or offered by each functional area.
- Identify the new internal and external customers that are recipients of the products and services of the functional area.
- Identify new providers and their products/services that are critical to the functional area.
- Relate new customers/providers to major functional areas through the products and services that are provided.

- Analyze the new functions of the organization that support the servicing of external customers to determine how this servicing can be provided at a lower cost or with more value.

3.4.3.2.4 Document Functional Area Relationships

The relationships, interactions, and dependencies between the new functional areas are specified as well as with existing functional areas. One method for performing this step is to construct usage scenarios that describe how certain business operations are performed. The new functions and entities are correlated with the business operations through the scenarios.

It is important to document and describe the activities of the organization's business (logical) and work units (physical).

One way to do this is through diagrams and matrixes that show the relationships of business units to each other and to the work units. The diagram should include a brief narrative and a glossary of the terms used. It should decomposed to no lower than three levels because it is intended to be correlated with the infrastructure descriptions developed in section 3.3.4.

3.4.3.2.5 Define/Update the Target Functional Architecture

This phase defines/updates the target functional architecture. It should include a description of the major functions—both automated and manual—performed by the business operations and the business applications which support them. The structure of the target functional architecture is the same as that of the baseline functional architecture that was depicted in appendix C.7.

3.4.3.2.6 Review the Target Functional Architecture

After the target functional architecture has been described, it should be reviewed with senior management and selected users and data owners to ensure a consistent, complete capture of the application resources and how they support the business functions.

The target functional architecture description should be circulated to the reviewers prior to meeting to review its contents. During the meeting major applications can be briefed to the reviewers. Comments from the reviewers should be used revise the target functional architecture description and the application definitions.

3.4.3.3 Define/Update Target Information Architecture

The purpose of this step is to document the major kinds of data and information needed by the organization and the interrelationships between the data and information, and to formulate a target CDM/DD TO BE used in formulating and planning the target functional architecture for the next iteration of the target architecture.

3.4.3.3.1 Define/Update Target Corporate Data Model and Data Dictionary

The ADP team defines/updates the target CDM/DD which reflect enterprise-wide data and information resources.

The target CDM/DD is a list of current and proposed data entities that are or will be used by the enterprise. The breadth and depth of this list (e.g., the amount of detail provided) depends on the number of data entities, the size and complexity of the enterprise, and the degree of detail required to begin the next iteration of the ADP. Examples of data entities include people, places, things, events, legislative changes, and regulatory provisions. Within the target CDM/DD, data entities are associated with business functions, business customers/providers (from the business view), and organizational work units (from the work architecture).

The ADP team should review the target CDM/DD descriptions and matrixes to determine whether redundant information is being retained within the organization. They should also review the entity relationship diagrams to determine whether overly complex or confusing data and information structures are being used. If so, they should determine whether some simplification of these structures is possible. The ADP team should also review new data entities to determine whether their definitions are consistent with other data entities. This information is not acted on immediately, but is presented to the senior management and the reviewers of the information architecture (see section 3.4.3.3.3) for decision.

3.4.3.3.2 Create/Revise the Target Information Architecture

After the target CDM/DD has been created along with the CRUD matrixes and the entity relationship diagrams, the target information architecture should be prepared which presents the current and proposed data and information entity definitions, their usage, and their relationships. Additional information that may be provided includes sizing and volume information, backup and recovery requirements, and integrity and security constraints.

The format of the target information architecture is the same as the baseline information architecture that is presented in appendix C.8.

3.4.3.3.3 Review the Target Information Architecture

After the target information architecture has been described, it should be reviewed with senior management and selected users and data owners to ensure a consistent, complete capture of the current and proposed data and information resources.

The target information architecture description should be circulated to the reviewers before the meeting for review. During the meeting the target CDM/DD can be briefed to the reviewers. Comments from the reviewers should be used to revise the target information architecture and the target CDM/DD.

3.4.4 DEFINE/UPDATE TARGET INFRASTRUCTURE

This step describes the computer systems that comprise the target infrastructure, which consists of three types of computer systems:

- Existing computer systems that are retained in the next generation of the architecture
- Proposed computer systems to be developed
- Existing computer systems to be eliminated or decommissioned as the next generation is built

3.4.4.1 Conduct Proposed Infrastructure Survey

This phase surveys the proposed infrastructure within the organization in order to create a target infrastructure inventory. The survey should focus on the same areas as identified in section 3.3.3.1.

3.4.4.2 Describe Proposed Infrastructure

Identify and specify the proposed computer system that supports a particular functional area. New functional areas may be supported by new computer systems as well as modified existing systems. One or more computer systems may support a functional area. The information to be provided has been enumerated in section 3.3.3.2.

3.4.4.3 Specify Interfaces between Infrastructure Components

Computer system interrelationships can be identified through a data flow and/or system diagrams. A data flow diagram documents the data that flows between computer systems either internal or external to the organization. This is not actually documenting the data flow, but showing the interrelationships, shareability, and accessibility of data in a process or activity.

As appropriate, develop graphical schematics of the relationships between current and proposed computer systems. Annotate the major information flows that occur among them. Matrixes similar to those developed in section 3.3.3.3 should also be developed.

3.4.4.4 Correlate Infrastructure with Business Operations

This phase correlates the infrastructure computer systems with the business operations that own and/or use them. Each new computer system is described to a detail sufficient to identify the major hardware, software, and telecommunication components; the major applications executed on the systems; and the physical and operation locations of the systems. Matrixes are prepared showing the relationship of new computer systems to business operations, new computer systems to applications, and new computer systems to physical locations.

3.4.4.5 Create/Update the Target Infrastructure

The target infrastructure describes the computer systems at the hardware and software platform level. It gathers information from the previous steps in a narrative or other descriptive form of the computer systems used in the organization.

A matrix mapping hardware and software platforms versus business applications should be created to show which business applications use which computer systems. This matrix will show the extent to which different types of technology are used throughout the organization.

The format of the target infrastructure description is similar to that of the baseline infrastructure description that is presented in appendix C.9.

3.4.4.6 Review the Target Infrastructure

After the target infrastructure has been described, it should be reviewed with senior management and selected users and data owners to ensure a consistent, complete capture of the hardware, software, and telecommunications resources. The target infrastructure should be circulated to the reviewers before the meeting for review. During the meeting the target infrastructure inventory can be briefed to the reviewers. Comments from the reviewers should be used to revise the target infrastructure and the target infrastructure inventory.

3.4.5 CREATE/UPDATE TARGET ARCHITECTURE DOCUMENT

The target architecture document includes

- Architectural overview
- Functional area descriptions
- Environmental description
- Key assumptions and rationales

The target architecture description must present possible solutions with sufficient clarity so that the customers can envision the proposed solution.

The suggested outline for the target architecture is presented in appendix C.12.

3.4.6 SUMMARY OF ACTIVITY 2

The steps, inputs, and outputs of Activity 2 are summarized in the following table.

Table 3-3. Activity 2 Summary

Activity	Inputs	Work Products	Deliverables
Define/update business requirements	TISAF Business requirements	Business requirements	None
Define/update target enterprise business view	TISAF	Target enterprise business view	None
Define/update target EISA views	TISAF Baseline work architecture Baseline functional architecture Baseline information architecture	Target work architecture Target functional architecture Target information architecture	None
Define/update target infrastructure	Baseline infrastructure	Computer system descriptions Target infrastructure	None
Create/update target architecture document	None	None	Target architecture

3.5 ACTIVITY 3—DEFINE/UPDATE ARCHITECTURE TRANSITION PLAN

An organization requires a well-thought-out strategy for transitioning from the baseline architecture to the next generation target architecture. This strategy must consider the current business culture, the available technology, and the current knowledge and skill sets of both business and information systems personnel. The transition should be viewed as not a computer system "conversion," but as a process of assimilation of new capabilities. The organization does not convert to a new system, but rather migrates to an enhanced, robust information system that leverages existing information assets.

Given the baseline architecture and the target architecture, an organization must plan the transition from the baseline architecture to the target architecture by developing an architecture transition plan. This planning process is performed in two phases: transition planning (this section) and implementation planning (see section 3.6).

The steps comprising the development of the architecture transition plan are depicted in figure 3-5. Each activity is discussed as if it were being performed for the first time. However, differences between the initial definition and the successive updates are explicitly noted.

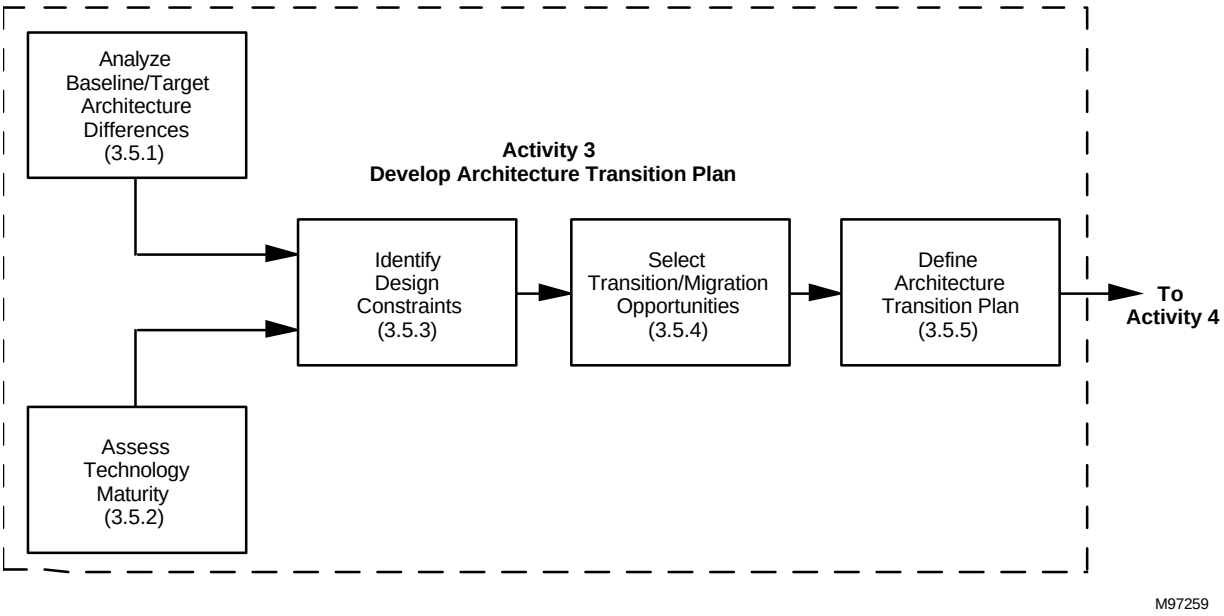

Figure 3-5. Defining/Updating the Architecture Transition Plan

3.5.1 ANALYZE BASELINE/TARGET ARCHITECTURE DIFFERENCES

The first step in architectural transition planning is to identify the differences between the baseline and target architectures, i.e., to perform a gap analysis. Gap analysis identifies the differences in the baseline and target architectures from the four architectural views: work, information, functional, and infrastructure. The target architecture has not been designed in isolation, so much of the analysis and many of the differences may have already been identified during the development process. The gap analysis also enumerates the components that need to be changed for each difference to be resolved. This step may be performed concurrently with the assessment of technology maturity (see section 3.5.2).

Gap analysis assesses the state of the legacy systems, e.g., those systems represented in the baseline architecture. It determines which baseline architecture components can be incorporated into the target architecture. This determination must balance the architectural and financial requirements for integrating legacy systems with new technology. A primary principle that must be applied is “sum costs don’t count.” Legacy systems only have value in terms of their future contributions, not in terms of their past costs.

The major work product of this step is the architecture difference description for which a template is provided in appendix C.13.

3.5.2 ASSESS TECHNOLOGY MATURITY

As part of the gap analysis, an assessment of the technological maturity of the baseline infrastructure and the required maturity of the target infrastructure is required. This step may be performed concurrently with the gap analysis of the baseline and target architecture (section 3.5.1).

Information resources can be described by their technological maturity. In some cases, the Department may be a technological leader, while in other cases the Department may lag behind the general marketplace in the deployment of technologies. In some cases, the Department wants to use state-of-the-art or even leading edge technology to accomplish its mission, but in other cases, the Department, of necessity, may need to take a more conservative approach.

The Department of Energy has defined five levels of technological maturity that can be used to assess the information resources described for an EISA (DOE 1996):

- *Experimental.* This level of maturity is reserved for initial technology examination, proof of principle pilot projects, and exploration of business benefits for a possible future architectural capability.

- *Adoption and expansion.* This level of maturity is reserved for deliberate, planned architecture project implementations, development, or deployments that add to or even completely replace existing baseline capabilities.

- *Formalization.* This level of maturity is reserved for deliberate, planned architecture projects that continue to add or replace existing baseline capabilities.

- *Mature.* This level is reserved for deliberate, planned architecture projects and other initiatives that continue to provide established and proven baseline capabilities.

- *Obsolete.* This maturity level is reserved for technologies and capabilities within the baseline, which are, or should be, considered for elimination or replacement.

The organization should assess each information system and its components according to the five levels given above. This information can be organized into a table for use in the transition planning process.

3.5.3 IDENTIFY DESIGN CONSTRAINTS

This step identifies and assesses the design constraints that may be placed on or by the organization in attempting to implement the target architecture. These constraints focus on availability and performance of technology, feasibility of approach given a particular technology, and interoperability of particular technology choices.

Constraints are factors that are unaffected by architectural changes. Constraints differ from requirements and goals in that they can involve factors from the larger world outside the organization as well as resource and technology constraints. Some examples of how constraints might be stated include:

- "How often" (functional/behavioral constraint)
- "How fast" (performance constraint)
- "How big"

- "How accurate"
- "How implemented" (physical constraint)
- "How delivered"
- "How it looks" (user interface constraint)
- "How it works" (operational constraint)

These constraints do not include budget, resource, and schedule limitations which are addressed in the implementation planning activity (section 3.6).

The results of this analysis are documented in a constraint summary report, which gathers together in one place the known constraints on the EISA and is included in the architecture difference document.

3.5.4 SELECT TRANSITION/MIGRATION OPPORTUNITIES

This step reviews the collection of gaps between the baseline architecture and the target architecture to determine which enhancements, revisions, or replacements should be accomplished in this iteration of the ADP. It applies the design constraints that have been identified to determine which opportunities are feasible within the next generation of the EISA. The results of this effort are a set of work products which are used to develop the architecture transition plan and memoranda describing the rationale for selecting/deselecting specific opportunities.

3.5.4.1 Identify Critical Differences

The set of differences identified during the gap analysis are classified as either critical or noncritical. Critical differences are those that will affect the successful accomplishment of the enterprise's mission. For example, a critical difference might be the implementation of new information systems and the modification of existing information systems to support a new business operation—perhaps in response to a legislative mandate. A critical difference might also include a "stay in business" transition to a new hardware/software platform running legacy applications. If the transition also entails the retirement of one or more existing systems, careful planning must occur to ensure that old systems are not retired before new systems are thoroughly tested, installed and certified as operational.

3.5.4.2 Develop Requirements/Constraints Decision Taxonomy

This step maps the constraints against the business, functional, and operational requirements in order to identify issues that require further analysis and resolution. Because one constraint may apply to many different business, functional, and operational requirements, a matrix is often used to present this information.

A taxonomy of requirements versus constraints is produced as a work product to assist decision makers.

3.5.4.3 Perform Feasibility Analysis and Resolve Design Issues

Every part of the EISA contributes to the solution, e.g., the support and sustenance of business operations. Similarly, every component, unfortunately, can also contribute error. A coordinated approach is required to ensure correctness, consistency, and completeness of the EISA. When errors cannot be prevented, it is less costly to fix them before they affect the implementation of the EISA.

3.5.4.4 Analyze and Assess Risks

Risks are potential problems that can prevent or impede the successful completion of a project. Mitigating risks throughout the life cycle of a project can ensure that such unwanted outcomes as late deliveries, cost overruns, or system failure do not occur.

During the evolution of existing information systems, risk can be introduced in two ways. First, new components must be designed to interoperate with existing components. Second, the new components must provide new functionality while continuing to perform retained existing functionality.

A risk assessment involves an evaluation of problems (risks) that might jeopardize successful completion of a project. Risks might include an insufficient skill base to do the work, insufficient funds to complete the project, inadequate training for users, and many others.

Risk assessments are cited as one of the best practices to use to ensure solid management of a large project. By effectively managing risks, project managers can greatly improve their ability to deliver projects on time and within budget and to meet or exceed objectives and customer expectations.

An assessment begins with the identification of potential risks to the project, based on experience with similar projects. The initial evaluation usually results in many risk factors being classified at an acceptable level. Those risks that appear to be at an unacceptable level are evaluated further. Abatement strategies may be proposed to address the unacceptable risks.

3.5.4.5 Identify "Quick Hit" Opportunities

"Quick Hit" opportunities are projects that can be delivered on a short-term payoff basis that yield new or enhanced functionality. It is anticipated that as these projects are completed in a relatively short time frame, they may be completed long before the next iteration of the ATP.

3.5.4.6 Describe Candidate Information Systems

Describe the proposed information systems that will support new functional areas. New functional areas may be supported by new information systems as well as modified existing systems.

Information system interrelationships can be identified through a data flow and/or system diagrams. A data flow diagram documents the data that flows between information systems either internal or external to the Department or organization. This is not actually documenting the data

flow, but showing the interrelationships, shareability, and accessibility of data in a process or activity. This information is very useful when making decisions regarding business applications.

Matrixes are one of the best tools for summarizing and assessing the criticality and usage and for providing a picture of the total environment. A distinction should be made between creation or modification of data elements and passive use (read-only), which can be accomplished using the Created-Reference-Update-Delete (CRUD) method. The information obtained in the subtasks above is used for the creation of the following three matrixes:

- *Data stores used by information systems.* This matrix shows which data stores are involved in which systems. The rows of the matrix list the data stores and the columns list the information systems. The data should be analyzed, using CRUD designations, according to its impact on the application. This will help to identify how and which systems can share and exchange data.

- *Subject areas (entity types) represented by data stores.* This matrix shows which subject areas (entity types) are represented by which data store in a similar manner to the matrix above. Not all data about an entity nor data about all entities of a given type need to be in the same data store. This matrix will quickly identify where an application should go to get the data it needs to perform a function, process, or activity.

- *Business functions supported by business applications.* This matrix shows which business functions are supported by which business applications. It is the most important matrix as it will determine whether current and proposed applications will meet the needs of the target work architecture and what is missing. It can also show duplications and where consolidations may be possible.

Ultimately, this information and these matrixes can be used to identify applications that create data, which should be implemented first, versus those that use data.

As appropriate, develop graphical schematics of the relationships between information systems. Annotate the major information flows that occur among them. Validate these graphical schematics with the entity relationship diagrams prepared in the functional architecture.

3.5.5 DEVELOP ARCHITECTURE TRANSITION PLAN

The architecture transition plan describes the milestones and activities, including transient systems, that must be performed to develop the target architecture from the existing baseline architecture.

3.5.5.1 Identify/Update Development Milestones

This step develops a milestone schedule for the major development necessary to transition from the baseline architecture to the target architecture. It establishes priorities for specific system development activities based on the interdependencies among the target information systems. The key principle to be adhered to is:

Develop, implement, and install information systems that create or collect data before information systems that use data.

This principle is extended to include the following:

- Develop transient systems that will serve as bridges between legacy systems and new systems before the new systems are developed.

- Develop software wrappers (e.g., new interface software) for information systems that will be retained for several iterations of the architecture before the systems accessing them are developed.

- Where possible, migrate existing legacy systems to interface with either transient systems or wrapped systems.

- Begin development of long-lead information systems that interface with only one or two other systems as early as possible (budget and resources permitting).

3.5.5.2 Assess Impact to Existing Systems

The purpose of this step is to determine the effects of implementing and installing components of the EISA on the baseline architecture. The correlation between business requirements and existing and planned information systems is reviewed. Each existing information system is classified as to whether it is to be

- Completely replaced by a new information system

- Partially replaced and/or modified through enhancements or modifications to its hardware and software

- Retained with minimal enhancements and modifications

The amount of time devoted to this impact analysis depends on the organization's schedule and available resources. Some of this impact analysis may be deferred until the system development life cycle has begun for information systems in this iteration of the EISA. Enough analysis should be conducted to identify those systems that are being retired because they are either being replaced or eliminated in order to allow adequate planning for their shutdown and removal.

3.5.5.3 Identify/Update Technology Acquisition Data

Once the milestones are identified, an analysis of the technology acquisition data can be prepared to identify those technology items (hardware, software, telecommunications) that need to be ordered because they either replace existing systems or are required to support the development of new software and capabilities. It is particularly important to identify long-lead items where there is substantial risk on delivery delays that can have a significant impact on the overall program schedule.

3.5.5.4 Develop/Revise Architecture Transition Plan

The ADP team develops and/or revises the architecture transition plan, which identifies the milestones for developing the next generation architecture, and identifies the changes that must be made to existing information systems.

A suggested outline for the architecture transition plan is presented in appendix C.14.

3.5.5.5 Review Architecture Transition Plan

The ADP team reviews the architecture transition plan with senior management, the sponsoring senior executive, and the stakeholders. Because the effects of the architecture transition plan extend beyond the information systems staff, a broad review across the organization and possibly including some external stakeholders is required.

3.5.6 SUMMARY OF ACTIVITY 3

The steps, inputs, and outputs of Activity 3 are summarized in table 3-4.

Table 3-4. Activity 3 Summary

Activity	Inputs	Work Products	Deliverables
Analyze baseline/target architecture differences	Baseline architecture Target architecture	Architecture difference document	None
Assess technology maturity	Baseline architecture Infrastructure inventory	Technology assessment report	None
Identify design constraints	Business requirements	Constraint summary report (included in architecture difference document)	None
Select transition and migration opportunities	Architecture difference document Baseline architecture Infrastructure inventory	Memorandum on “quick hit” opportunities	None
Define/update architecture transition plan	None	Milestones list	Architecture transition plan

3.6 ACTIVITY 4—DEFINE/UPDATE ARCHITECTURE IMPLEMENTATION PLAN

Once an architecture transition plan has been developed, the implementation planning for the target architecture begins. This activity identifies, assesses, and imposes budget, resource, and schedule constraints on the elements of the architecture transition plan. As a result, control flow may return to Activity 3 for replanning based on inconsistencies or problems identified during this activity.

The major deliverable of this activity is a revised architecture transition plan with additional information regarding specific projects, budget and resource allocations, and schedules identified.

The steps comprising the development of the baseline architecture are depicted in figure 3-6. Each activity is discussed as if it were being performed for the first time. However, differences between the initial definition and the successive updates are explicitly noted.

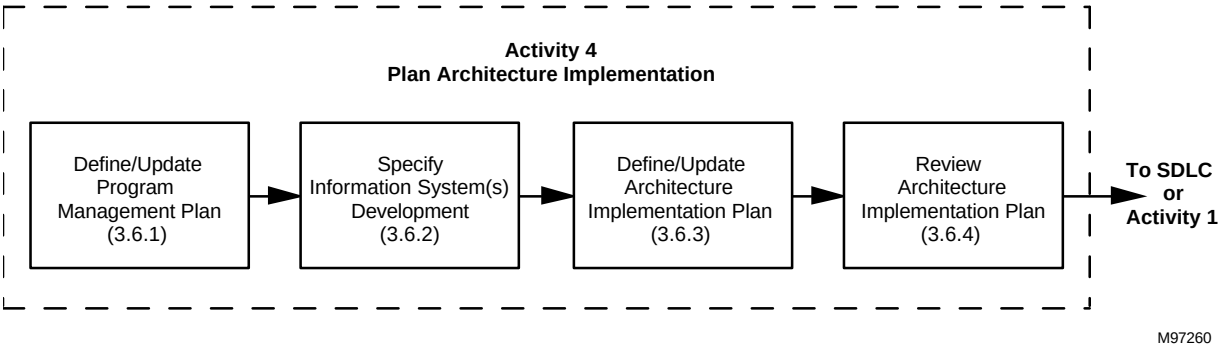

Figure 3-6. Defining/Updating Architecture Implementation Plan

3.6.1 DEFINE/UPDATE PROGRAM MANAGEMENT PLAN

The migration to a target architecture should be the result of a carefully planned program of activities with milestones and checkpoints that measure progress, assess risk, and enable corrective actions, when necessary.

A program management plan for implementing the target architecture is a subset of the overall architecture transition plan. It focuses on budget, schedule, personnel, and technology.

The following steps are performed in parallel and iteratively. A single description of this data usually does not emerge during the first iteration. During subsequent iterations, the initial estimates and allocations are refined and honed as the interaction between different elements becomes clearer.

3.6.1.1 Estimate Information Systems Develop Time

The architecture transition plan may not specify which components of information systems software are to be purchased unless there is a clear and compelling business or technical requirement to do so. In preparation for resource assessment and allocation, a preliminary determination of the development time required for software components is performed. At the same time, candidate software packages can be reviewed and assessed for their ability to meet the business requirements. Specific software packages that are available from vendors that are likely to meet the business requirements may be identified. These will be used to ascertain that commercial software is available and get an estimate of relative costs to acquire.

The lifetime costs to purchase and modify or integrate a commercial software package may exceed those of internally developing software to provide the requisite capabilities. However,

other factors such as resources available, total budget, and schedule must be evaluated in a tradeoff analysis.

For each information systems component that may be implemented by purchasing a software package, an estimate is made of the time and resources required to evaluate, select and acquire the package; install it; modify the dependent software; and place it into production. Since one package may satisfy multiple requirements, the effect on the overall implementation sequence should be evaluated.

3.6.1.2 Estimate Resource Requirements

Estimates of the resources required to develop information systems are developed. The estimate should take into account

- Personnel availability (various skills categories)
- Software tools and technology platforms availability
- Facilities availability and locations where development is to be performed
- Data sources available to support in test and evaluation

Depending on the time available to perform the estimation process, one or more of the following alternatives may be considered for developing the estimates (Spewak 1993, 255):

- Rely on the experience and expertise of the architecture development team.

- Rely on the advice of consultants and other system scheduling experts, including academic experts.

- Use software packages that have been developed for making such estimates.

Using estimating packages may provide two important benefits. First, because the estimation, scheduling, and allocation effort is a tedious, time-consuming activity, such products yield a database that can be used be individual system development projects as well as allowing the evaluation of what-if alternatives and full-life-cycle costing. Second, most estimation packages provide validation and cross-checking features that can improve the quality and credibility of the estimation process.

3.6.2 SPECIFY INFORMATION SYSTEM DEVELOPMENT

This step is performed for each new and existing information system identified in the target architecture. Modifications can affect any aspect of the information system. Specifically, modifications can be required for the functions, information or data, or the technologies that provide the infrastructure for the information system. Each modification must be justified in that it improves the support to the day to day operations of the enterprise in some specific way.

Modifications to functions can require changes to the input source, process, output destination and the organizational entities responsible for it. Data modifications can be in the form or input

format, data definition and/or data management, and the output format. Changes to the technical infrastructure can affect the hardware, operating system, supporting system software, communications or other peripheral devices, and the physical location for execution of the function. The change should be presented with respect to the original function, data or technical infrastructure if one existed. In addition, side effects to other functions, data or technical infrastructure should be described.

The rationale for each change must be provided. In addition, any constraints, anticipated resource requirements and other impacts should be described.

3.6.3 DEFINE ARCHITECTURE IMPLEMENTATION PLAN

The resource estimates, system interdependencies, and system development cost estimates are combined to produce an implementation schedule. This schedule identifies the specific system releases (correlated with the milestones), the start and end dates for specific activities, the elapsed time, and total effort required to develop and deliver each information system.

The length of the implementation schedule is based on the needs of the organization and the direction of the iteration of the architecture development process. Because some projects are likely to slip, it is worthwhile to consider at least two iterations of the architecture development process (with the second iteration not as well-defined) in order to determine how project slippages will impact the overall schedule.

The implementation schedule allows the architecture development team to analyze resource usage, identify conflicts in personnel, facilities, and individual information system schedules, and resolve these through project slippage or resource allocation enhancements.

A suggested architecture implementation plan is presented in appendix C.15.

3.6.4 REVIEW ARCHITECTURE IMPLEMENTATION PLAN

When the previous steps are completed, the architecture implementation plan is reviewed to reflect the decisions reached with regard to resources and technology. During this review, some elements of the target architecture may need to be modified or deferred to reflect the changes in resources, availability of personnel, or restrictions on budget or schedule.

3.6.5 SUMMARY OF ACTIVITY 4

The steps, inputs, and outputs of Activity 4 are summarized in table 3-5.

Table 3-5. Activity 4 Summary

Activity	Inputs	Work Products	Deliverables
Define/update program management plan	Existing program management plan Budget projections	Program management plan	None

Specify information system development	Relevant information systems descriptions	Implementation schedule	None
Define architecture implementation plan	Implementation schedule	Architecture implementation plan	None
Review architecture implementation plan	Architecture transition plan Architecture implementation plan	None	Revised architecture transition plan Revised architecture implementation plan

3.7 ACTIVITY 5—ADMINISTER ARCHITECTURE DEVELOPMENT PROCESS

Developing information system architectures that reflect a common purpose and incorporate standards embraced across an organization requires a support system dedicated to maintenance of the quality of the architecture development process. There must be a way to review the architecture development process to ensure that standards are followed, that new standards are reviewed and incorporated into the process and outdated standards deleted when appropriate. Architectures must be reviewed to ensure quality and to be sure that the process meets the needs of the organizations charged with its implementation. For the TISAF, that responsibility rests with the Treasury Architecture Working Group (TAWG).

3.7.1 ARCHITECTURE WORKING GROUP

The architecture development process must be continually refined to respond to the changing environment in which the Treasury bureaus operate. The responsibility for managing and promoting the TISAF and the Treasury Architecture Development Guidance (TADG) resides in the enterprise's Architecture Working Group (AWG). At the Departmental level, this responsibility resides in the Treasury Architecture Working Group (TAWG), a working group of the Treasury CIO Council.

The responsibilities of the AWG include

- Overseeing the development and maintenance of the architecture development process, ensuring representation of departmental academic and administrative requirements.
- Maintaining the baseline and target architectures and supporting documents
- Documenting and publicizing architectural decisions rendered by the CIO Council
- Maintaining current awareness of external standards issues appropriate to the organization, but consonant with the TISAF standard profile

3.7.2 PROVIDE/MAINTAIN PERSONNEL AND SKILLS

Architectural development is a people process. People make the decisions that the architectural descriptions and the architectural transition plan contains. The ADP plan lists the people (by skill category) whose involvement will be necessary. Once specific team members are assigned, they

can determine from the ADP plan and its lists of steps how much time they need to allocate to the effort.

A critical element of any architectural development project is the acquisition and/or retention of qualified, skilled personnel. This requires that Treasury be able to identify qualified staff before they are hired and that they train existing employees to improve and maintain their skills.

3.7.2.1 Acquiring Skilled Personnel

It is essential that staff skilled in the technical areas and topics addressed by the TISAF and TADG be available to support the Treasury organizations in implementing the architecture development process. One way to ensure this skill base is available is to acquire new staff with the requisite skills. This step should be done as early as possible in the architecture development process and continues throughout the lifetime of the enterprise's architecture development activities.

A set of criteria must be developed that reflects the skills necessary to support the implementation of the TISAF and the TADG. Once the criteria are defined, applicants can be evaluated against the criteria to determine the degree to which they match the required skill set. This criteria must be reviewed and updated periodically to ensure that the skills required and the ones reflected in the criteria are consistent.

Skills that should be included reflect the goals and objectives of the TISAF and its design guidance, such as skills with

- Architecture
- Hardware
- Software
- System engineering
- System design
- System integration
- User interface
- Quality assurance

Specific criteria associated with each skill category should be defined to support more refined assessment of candidate personnel.

3.7.2.2 Providing Training

Once staff have been acquired, their skills must be maintained or they will become obsolete. As new standards, techniques and technologies are identified for incorporation into the architectures, courses must be identified to ensure staff are trained in the newest techniques. These course may be available outside Treasury, or they may be imported from outside vendors or produced in-house. The list of courses must be reviewed periodically to keep it current and ensure that the right courses are offered.

In addition, a database must be created to track employee training. A history should be kept of all courses employees have taken along with their dates and times.

3.7.3 PROVIDE/MAINTAIN ARCHITECTURE DEVELOPMENT TOOLKIT

An architecture development toolkit provides a set of tools for supporting the architecture development process. Any acceptable means for modeling business processes and information flows may be used. Since there is no single method of modeling systems and information architectures for use within the Treasury Department or the government, Departmental components can use commercial computer-aided software engineering (CASE) tools. The tools should be selected on the basis of the level of prior experience with information modeling, size and strategic importance of the project, prior investments in place, and the ability to reuse analytical products, meta-data, data structures, and generated code, and other considerations.

Development and documentation of information architecture designs with automated tools is advisable to merely intuitive or academic alternatives. Object-oriented tools are more useful than entity-relationship modeling which provides less insight into meta-data needed by other users. However, entity-relationship modeling is better than the absence of modeling of any kind. The existence of tools and/or capital infrastructure tailored to particular architectures will exert a bias towards those architectures.

3.7.4 PROVIDE/MAINTAIN ARCHITECTURE DEVELOPMENT LIBRARY

As the EISA is developed, numerous documents providing information about the enterprise or departmental components are consulted. These documents should be cataloged and filed in a central location for continuing and future reference. During successive iterations of the methodology, the architecture reference library should be maintained through addition of new and/or updated documents from departmental components.

3.7.5 DEFINE/REFINE CRITICAL SUCCESS FACTORS

An effective architecture development program requires that measurement of important business activities and information systems services be conducted prior to and after initial investments are made. Measuring of information resource services and information technology performance are essential means to evaluate comparative costs, benefits, and effectiveness resulting from architecture design decisions. Departmental components are strongly encouraged to establish and maintain information resource measurement capabilities and to periodically review ISA usage patterns. All Departmental components are also strongly encouraged to adopt measurement approaches and indicators that enable comparisons to other Departmental business units, federal and state agencies, and the experience of leaders in the commercial sector.

3.7.6 SUMMARY OF ACTIVITY 5

The steps, inputs, and outputs of Activity 5 are summarized in table 3-6.

Table 3-6. Activity 5 Summary

Activity	Inputs	Work Products	Deliverables
Architecture Working Group	TISAF	None	None
Provide/maintain personnel and skills	None	None	None
Provide/maintain architecture development toolkit	Existing tools suite	Architectural development tools suite	None
Provide/maintain architecture development library	Source documents Work products Major deliverables	Architecture development library	None
Define/refine critical success factors	TISAF	Critical success factors	None

3.8 APPLYING THE ARCHITECTURE DEVELOPMENT PROCESS

The architecture development process described in this section can be adapted and customized by Departmental components to meet their needs for defining and documenting their baseline and target architectures and developing an architecture transition plan. Individual components are not required to use this process to develop their architectural description documents.

This subsection provides some additional guidelines for applying the architecture development process to meet a Departmental component's needs.

3.8.1 GENERAL ADP GUIDELINES

The following guidelines are suggested for consideration by departmental components when applying the ADP:

- Do not table a critical/damaging issue/concern/problem too far into the process, but resolve it as quickly as possible.

- An architecture development project should not begin without senior management sponsorship and representation on the ADP team.

- Assign as the ADP team leader a strong effective leader who has the ability to overcome unforeseen problems and provide strong, focused leadership.

- Assign as ADP team members personnel who are knowledgeable of the organization where they work, because they will have the best understanding of how the organization operates and what its needs are.

- Appoint a technical group which includes subject matter experts to participate in the ADP

- Select an appropriate methodology for performing the analysis in the subsequent activities of the architecture development process. These may be selected by the project leader or by the project team.

- If an organization is inexperienced in architecture development, consider using a consultant to assist in the execution of the architecture development process, including facilitating the ADP team meetings.

- Enterprise-wide problems must be solved with enterprise-wide solutions.

- An architecture must be developed according to the unique business and information environment of the Department.

- As practical guidance, the development of an EISA, whether baseline or target, should not take more than four to six months during the first iteration. However, this estimate must be mediated by the size of the organization, the availability of information, and the detail required by the senior management.

- Plan on spending more time and resources the first time the architecture development process is executed because there is no work to build on. Succeeding iterations should require less time to complete because they build on and extend work that has already been done.

- Information which is classified as Department-level must be made available by the data owners across the Departmental infrastructure, taking into account the appropriate security concerns.

- Department-level information may be duplicated in several components; efforts to coordinate the data definitions will require close cooperation among the components to develop appropriate standards.

- Archiving and records management considerations must be taken into account during the architecture development process.

- It is recommended that once an architecture development process is begun, it should be carried out through one complete iteration.

3.8.2 BASELINE ARCHITECTURE GUIDELINES

The following guidelines are suggested for consideration by Departmental components when developing the baseline architecture:

- Ensure that a good data modeler participates in the information architecture step.

- Ensure that the modeling methodologies for the different EISA views can form a single consistent view.

- Users get information to support business processes by accessing various applications.

- Applications and data function with the technology architecture through modularly designed standard interfaces. Individual applications take advantage of the infrastructure to interoperate and access data.

- An adaptable infrastructure provides the capability to add onto the current investment with minimum inconvenience to the user.

- Adaptability and life expectancy are major criteria in setting infrastructure standards and selecting components.

3.8.3 TARGET ARCHITECTURE GUIDELINES

The following guidelines are suggested for consideration by Departmental components when developing the target architecture:

- When developing a target EISA, a target date should be specified; i.e., "the target architecture will be accomplished by the year xxxx." The vision year should extend at least several years from the current year.

- Ensure that component-level target architectures are coordinated with and checked against Department IT initiatives.

- Ultimately, the vision owners for the target architectures are the business operations and work units; they should be the consulted on all elements of the target architecture.

- Ensure that a good data modeler participates in the information architecture step.

- Use the same data modeling methodology for the target information architecture as for the baseline information architecture.

- Ensure that the modeling methodologies for the different EISA views can form a single consistent view.

- Place priority on systems and applications that support strategic business initiatives.

- Develop a technical and organizational information systems infrastructure capable of meeting current and future needs.

- Organizational requirements are both unique and dynamic. The infrastructure should support an environment that allows new applications to start small, quickly, and inexpensively.

- Adaptability and life expectancy are major criteria in setting infrastructure standards and selecting components.

- Architectures that demonstrate scalability and adaptability and support portability are more likely to be maintained or supported in the future.

- Architectural solutions must account for control agency statutory responsibilities for data validity even though the data may be collected elsewhere.

3.8.4 ARCHITECTURE TRANSITION PLAN GUIDELINES

The following guidelines are suggested for consideration by departmental components when developing the architecture transition plan and revising it with implementation data:

- The architecture transition plan should be composed of phases which can be accomplished in increments and yield intermediate capabilities.

- Customers control the allocation of critical resources.

3.8.5 ADP ADMINISTRATION GUIDELINES

The following guidelines are suggested for consideration by Departmental components when developing mechanisms for administering the architecture development process:

- System architects and users will need to be trained to challenge assumptions requiring the capture of information and to test these against current and future needs.

- ADP members need to be trained in archiving and records management policies and procedures in order to meet legislative and Departmental mandates.

- ADP managers are responsible for assuring that information is protected and that controls are in place to ensure that the information is being collected and used properly in the architecture development process.

- A policy is required to distinguish invalidated raw data, notes, working papers, etc. from official data and information.

APPENDIX A
DEFINITIONS AND ACRONYMS

Access Certificate: An access certificate is like a ticket which grants access to certain services and may be passed from one client to another (Buill and Herbert 1994).

Access Control: The process of limiting access to the resources of a system only to authorized programs, processes, or other systems (in a network).

Access Control List (ACL): An ACL is a table that maps client identities against access rights to available services. If a client requests a service from a server, the server checks the authority of the client to make the request by reference to an ACL (Bull and Herbert 1994).

ADD: Architecture Difference Document

ADL: see Architecture Description Language

Adoption: A decision to make full use of an innovation or information technology (Rogers 1982).

ADP: Architecture Development Process

ADT: Architecture Development Team

Applet: A small Java program that can be embedded in an HTML page. Applets differ from full-fledged Java applications in that they are not allowed to access certain resources on the local computer, such as files and serial devices, and are restricted to accessing a limited set of other computers networked resources. The current rule is that an applet can only access the computer from which it was came from.

Application: A computer program designed to perform a specific function or process for a user. An application is generally used by itself, but may be designed to exchange information with other applications, thus making it sharable. Sharable applications are called services.

Application Engineering: The process of instantiating, refining, and/or extending a reference architecture to develop one or more applications.

Application Programming Interface (API): A set of formalized software calls made available by one program or objects for the benefit of other programs or objects.

Architecture Description Language (ADL): A set of languages, standards, and conventions for the models used to represent a DSSA. ADLs are intended to be used for the development of new systems based on existing architectures. Although denoted as “language,” an ADL may be either graphic or textual in nature (Lockheed-Martin 1997b).

Asynchronous Transfer Mode: A transfer mode in which the information is organized into cells, it is asynchronous in the sense that the recurrence of cells containing information from an individual user is not necessarily periodic. A flexible technology that enables a network to carry many types of traffic such as audio, video, and data, while providing enough bandwidth for each type of traffic and guaranteeing timely delivery of time-sensitive traffic. ATM is a cell-relay, connection-oriented, switched protocol for information transfer (http://titan.mic.dundee.ac.uk/glossary/telecomm/index.htm).

ATM: Asynchronous Transfer Mode

ATP: Architecture Transition Plan

Availability: The property threatened by denial of service.

BA: Baseline Architecture (see section 3.3)

Baseline: A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures (IEEE 610:1991).

Business Function: Group of business activities which together completely support one aspect of furthering the mission of the organization describing what is done within the organization independently from the organization structure.

Business Process: (1) Defined business activity, executions of which may be identified in terms of the input and/or output of entities of specific types or of data about entities of specific types—a process can be executed; a function cannot. (2) An ordering of work activities across time and place, with a beginning, an end, and clearly identified inputs and outputs (http://amex.cox.smu.edu:80/mis/talks/emba/bpr/process.html).

Business Requirement : A description of the kind of information needed, such as a report or statistics, to accomplish a business or organizational objective. It is synonymous with business need or information need.

Cache: A temporary fast storage area for data which would normally be accessed from a slower storage device. A cache management algorithm monitors the data access patterns and selects which data from the slower device is to be kept in the cache for quick access. Caches are normally transparent or hidden from the accessing device (http://www.dpt.com/glossary.html#A).

CD-ROM: (Compact Disk Read-Only Memory) A read-only storage device which retrieves up to 640 Mbytes of information from a removable laser disk similar to an audio compact disk.

Certificate: An electronic identifier from a certificate authority that includes a signature made by a Certification Authority with a private key. The authenticity of the signature is validated by other users who trust the Certification Authority’s public key.

Certification Authority (CA): A trusted authority responsible for issuing certificates used to identify a community of individuals, systems or other entities which make use of a computer network.

CGI: Common Gateway Interface

Client: A software program that is used to contact and obtain data from a server software program on another computer, often across a great distance. Each client program is designed to work with one or more specific kinds of server programs, and each server requires a specific kind of client. A Web browser is a specific kind of client (Internet Literary Consultants, http://www.matisse.net/files/glossary.html).

Client/Server Architecture: (1) A model of computing in which two programs cooperate to do work. The program that initiates the work is the client, which makes requests to the second program, the server, who executes them and returns the results. (2) A means of defining and integrating the components of an organization's technology infrastructure to enable an employee to gain appropriate access to data and systems needed to do their job, from a single desktop device, regardless of where the data and systems reside.

Cohesion: Cohesion is the measure of the strength of functional relatedness of elements (instructions, group of instructions, data definitions, function call) within a module. There are 10 types of cohesion, also called strength of cohesiveness:

- Module independence
- Module strength
- Function of a module
- Coincidental strength
- Logical strength
- Classical strength
- Procedural strength
- Communicational strength
- Functional strength
- Informational strength

Designers should create strong, highly cohesive modules. On the other hand, the elements of one module should not be strongly related to the elements of another module because it would lead to tight coupling between the modules. The notion of cohesion allows characterizations of the links between the different objects of a module (http://irb.cs.uni-magdeburg.de/se/explain/complexity.html).

Commerce Server: A commerce server is a server configured with appropriate security software and protocols, such as SSL, to accept credit card numbers and other sensitive information.

Communications Transparency: Hiding the complexities of multiple protocols and dissimilar data representations behind a set of abstractions for interprocess communication (Internet Literary Consultants, http://www.matisse.net/files/glossary.html).

Component: (1) An abstractly characterized unit that can provide capabilities or services to others. Generic components in a reference architecture may be parameterized or otherwise imprecise. Implementation components, on the other hand, are executable and generally have a physical embodiment in software or hardware or a mixture of the two. (2) One of the parts that make up a system; a component may be hardware or software and may be subdivided into other components (IEEE 610:1991).

Confidentiality: The property that information is not made available or disclosed to unauthorized individuals, entities, or processes. Data which reveals no information may be revealed.

Connectionless Service: A service which allows the transfer of information among service users without the need for end-to-end call establishment procedures.

CONOPS: Concept of Operations

Cooperative Processing: (1) Computing which requires two or more distinct processors to complete a single transaction (http://www.abs.net/~lloyd/csfaq.txt). (2) Client/server configuration in which application components are distributed across different processors.

CORBA: Common Object Request Broker Architecture (TISAF, D.2.1.3). A standard for ORBs defined by a number of IT companies on behalf of the Object Management Group which provides interoperability between ORBs.

Coupling: The strength connection between modules (software components). Therefore, it depicts a factor of intermodular complexity. Coupling is defined for modules that share data or have a control connection, which restricts the use to modules which share an executable path. However, other types of connections between modules which do not involve an executable path are possible. The following types of coupling may be distinguished (http://irb.cs.uni-magdeburg.de/se/explain/complexity.html).

- *Content coupling* between modules wherein one module directly references the code of another module.

- *Common coupling* between modules wherein two modules share global data structures.

- *External coupling* between modules wherein two modules only share a single variable (weak common coupling).

- *Control coupling* between modules where one module explicitly controls the function, i.e. I/O operation, of another module.

- *Stamp coupling* between two modules which reference the same nonglobal data structure.

- *Data coupling* between two modules where one module calls the other, and all inputs to and outputs from the called module are data item parameters (not structures).

Critical Success Factor: Identification of a performance measure which must be achieved if the organization is to succeed in its environment. The critical success factors are the limited number of areas in which satisfactory results will ensure the success of the project. Careful attention to and management of these factors by project management is essential to project success. There are different sets of critical success factors for the product of a project and for the project itself. Use critical success factors for the product of the project, unless the discussion is about the project management plan.

CRUD: Create-Read-Update-Delete—the four basic processes that may be performed on any data element. It is used to development a CRUD matrix which shows which operations may be performed upon which data.

Data Mining: The extraction of implicit, previously unknown, and potentially useful information from data; also known as "knowledge discovery in databases."

Data Warehouse: A data warehouse is an organized store of data designed to support decision making. The data in a data warehouse are

- Oriented to subjects, not processes
- Integrated, usually from several different applications or computer platforms
- Time-variant, that is, a snapshot of the data at a point in time
- Nonvolatile, that is the data is uploaded periodically and accessed, but not "updated"

(Minnesota Dept. of Administration, Information Policy Office)

DBMS: Database Management System. The software that handles database access requests from application processes.

DCOM: Distributed Common Object Model

Department-wide: Scope that is larger than one particular development project. Scope may be less than the full Department, but the techniques are vastly different from those to derive requirements for a single application or business system. This implies more emphasis on objectives and processes of the Department, rather on the requirements of individual users.

Digital Signature: The use of public key cryptography to assure the identity tied to data sent over a network. Digital signatures are created using a mathematical summary of the data to be sent (a hash), which is encrypted with the sender's private key and read by the sender's public key.

Distributed Availability: The ability to ensure predictable, reliable user access to key applications and computing resources.

Distributed Database: A database that consists of two or more data files located at different sites on a computer network. Because the database is distributed, different users can access it without interfering with one another. However, the DBMS must periodically synchronize the scattered databases to make sure that they all have consistent data.

Domain: A problem or task area in which multiple highly similar application systems will be developed to meet the particular requirements of several different customers.

Domain Model: (1) The terminology and semantics characterizing elements and relationships in a domain. The domain model defines the terms used to express requirements and evaluate systems. (2) A representation of a process—that is, what is happening in the domain. It tells what functions are being performed and what data, information, and entities are flowing among those functions. It is a representation of the behavior of all the entities in the domain. The behavior is a collection of functions and flow among functions.

Domain Modeling: An emerging class of software design methodologies that formally define families of related systems.

Domain-Specific Software Architecture (DSSA): An architectural specification specialized for a given domain, but generalized enough to construct a range of systems across the domain. A DSSA consists of a reference architecture, a domain model, and a set of reference requirements (Lockheed-Martin 1997b).

DSOM: Distributed Systems Object Model. A complete implementation of CORBA that was developed by IBM.

EISA: Enterprise Information System Architecture

Electronic Mail: The transmission of documents, notes, and messages across a network.

Encapsulation: The process of establishing a boundary around a object such that it is capable of functioning independently from other entities. Data and processing are “hidden” inside the object.

Ethernet: A local area network standard developed by Xerox, DEC, and Intel. Represented by IEEE 802.3 (TISAF, D.3.3.6).

Feature: A prominent or distinctive, user-visible quality or characteristic of a system.

Firewall: A protective barrier that is a collection of components configured to enforce a specific access control policy between a trusted network and an untrusted network.

4GL: Fourth Generation Language

FTP: File Transfer Protocol

Function: A major work element to accomplish the mission or business of an organization, such as accounting, marketing, etc. A *subfunction* is defined as a component of a function such as accounts receivable, accounts payable, etc. in the accounting function.

Gateway: Connection between two or more different networks (e.g., packet switched and circuit switched).

Geographic Information Systems (GIS): GISs combine the use of geographic information with descriptive information to enable spatial presentation and analysis.

Goal: A desired or needed result to be achieved by an agency over the long term. It may not be possible for an agency to achieve all of its goals within the planning horizon. Goals support the agency mission. They generally identify how the mission will be carried out, reflect the style of the agency, and project the public image to be created and maintained.

GOSIP (Government Open Systems Interconnection Profile): A procurement specification covering OSI-based data communications.

GUI (Graphical User Interface): A pictorial way of representing the capabilities of a system and the work being done on it.

Heterogeneous Environments: Systems that are not alike in structure, composition and/or function.

Homogeneous Environments: Systems that are alike in structure, composition and function.

HTML (Hypertext Markup Language): The coding language used to create Hypertext documents for use on the World Wide Web. HTML looks a lot like old-fashioned typesetting code, where a block of text is surrounded with codes that indicate how it should appear. Additionally, in HTML you can specify that a block of text, or a word, is linked to another file on the Internet. HTML files are meant to be viewed using a World Wide Web Client Program, such as Netscape or Mosaic (Internet Literary Consultants, http://www.matisse.net/files/glossary.html).

HTTP (Hypertext Transport Protocol): The protocol for moving hypertext files across the Internet. Requires a HTTP client program on one end, and an HTTP server program on the other end. HTTP is the most important protocol used in the World Wide Web.

Hypertext: Generally, any text that contains links to other documents—words or phrases in the document that can be chosen by a reader and which cause another document to be retrieved and displayed.

IDL: Interface Definition Language

IETF: Internet Engineering Task Force

IIOP: Internet Interoperable ORB Protocol

IMAP: Internet Mail Access Protocol

Information Management: The planning, budgeting, manipulating, and controlling of information throughout its life-cycle as defined by OMB Circular A-130

Information System Architecture: A conceptual and coherent blueprint that describes the structure of information system components, their interrelationships, and the architectural principles and guidance governing their design and evolution over time in an organization.

Information Technology: Includes equipment, software, services, and products used in processing information, office automation, and telecommunications.

Information Technology Project (IT Project): An effort having a planned beginning and end, with objectives that involve one or more of the following activities: studies, acquisition, design, construction, testing, documenting, installation, implementation, and operation of information technology resources.

INP: Intelligent Network Printer

Integrity: The property that data has not been exposed to accidental or malicious alteration or destruction (NCSC 1987).

Internet: The Internet is the international network of data networks, including research and education networks as well as commercial and other networks. The Internet is made up of local area networks, metropolitan area networks and wide area networks that are connected through "gateways" using the TCP/IP protocols. Used by many different communities in support of collaboration, cooperation and dissemination of information, the Internet is viewed by its creators as a public resource.

Internet Engineering Task Force: A nonprofit group of individuals who establish policy for the Internet.

Interoperability: The capability to allow users to readily share data among applications residing on varying combinations of hardware and software within and between existing networks.

Intranet: An intranet is a private corporate network based on Internet standards and technology.

ISDN: Integrated Services Digital Network, a digital telecommunications network built to an international set of standards that aims to carry information in a fast and error free manner.

ISP: Internet Service Provider

Java: A platform-independent object-oriented 3GL first developed by Sun Microsystems.

Legacy system: An automated system built with older technology that may be unstructured, lacking in modularity, documentation and even source code.

Local Area Network (LAN): A user-owned and operated data transmission facility connecting a number of communicating devices (e.g., computers, terminals, printers, storage devices) within a single building or campus of buildings.

LDAP: Lightweight Directory Access Protocol

Location Transparency: Users, services, and resources join and leave the network constantly, but they are never tied to fixed locations (BYTE Magazine, http://www.byte.com/art/9504/sec11/art1.htm#3waves).

Megaprogramming: The practice of building and evolving computer software component by component.

Middleware: Middleware is systems integration software for distributed processing and for database and user interfaces.

Mission: General statement of the purpose and nature of the organization. (The reasons to be in business, why the organization exists.)

Namespace Transparency: Everything on the global network must appear to belong to the same namespace. Names must resolve uniquely within a given context or naming authority (Byte Magazine, http://www.byte.com/art/9504/sec11/art1.htm#3waves).

Network Computer: A class of computers, possibly diskless, that are designed to run software written in Java (Oracle/Sun Microsystems trademark definitions).

NIC: Network Interface Card

NOS: Network Operating System

Objective: A measurable result, not an activity, that management has agreed upon to accomplish within a specific time frame. Strategic objectives generally have time frames of 5 to 10 years.

Office Automation: Application of information technology to increase the effectiveness of managerial, professional, and/or clerical workers performing office functions.

OLAP: On-Line Analytical Processing (OLAP)—A category of software technology that enables analysts, managers and executives to gain insight into data through fast, consistent, interactive access to a wide variety of possible views of information that has been transformed from raw data to reflect the real dimensionality of the enterprise as understood by the user (OLAP Council) OLAP functionality is characterized by dynamic multidimensional analysis of consolidated enterprise data supporting end user analytical and navigational activities including

- Calculations and modeling applied across dimensions, through hierarchies and/or across members
- Trend analysis over sequential time periods
- Slicing subsets for on-screen viewing
- Drill-down to deeper levels of consolidation

- Reach-through to underlying detail data

- Rotation to new dimensional comparisons in the viewing area

More on OLAP can be found at: (Datamation, http://www.datamation.com/PlugIn/workbench/olap/stories/white.htm).

OLTP: On-line Transaction Processing

OMG: Object Management Group

OOUI: Object-Oriented User Interface, a variant of a GUI.

Open System: (1) Software environments consisting of products and technologies which are designed and implemented in accordance with standards—established and de facto—that are vendor independent and commonly available (Hewlett-Packard 1996) (2) An information processing environment based on application programming interfaces and communication protocol standards which transcend any one vendor and have significant enough vendor product investment to assure both multiple platform availability and industry sustainability (Gartner 1991).

ORB (Object Request Broker): A mechanism by which objects make and receive requests and responses from each other, without needing to specify their location. ORBs provide interoperability among applications on different machines in heterogeneous, distributed environments through the interconnection of multiple object systems.

OS: Operating System

OSI: Open Systems Interconnect. The OSI 7-Layer model is used to define standards for all levels of communication. Each layer is aware only of the level above and below it. It takes information from the layer above, adds control information to it, and passes it to the layer below.

The corresponding layer on the receiving end takes information from the layer below, removes the control information, and passes it to the layer above. The seven layers are:

- Physical layer bitstream
- Data link layer error checking
- Network Layer routing
- Transport layer hides network from the session layer
- Session layer process-to-process communication
- Presentation layer text handling
- Application layer what the user sees

PDD: Presidential Decision Directive

Proof of concept: A working model that is used to validate and explore key risks or unknowns within the architecture.

Protocol Engineering: The practice of planning, designing, and implementing network and WAN protocols in order to optimize network performance

Proxy Server: A special-purpose web server that (1) replicates or caches pages from other web servers (usually behind a firewall) that are to be made publicly available, and (2) protects web servers holding sensitive information by filtering the packets that can be sent to the web server.

Public Key Certificate: The credentials used to authenticate Web servers and their clients via protocols such as SSL and S-HTTP are called X.509 public key certificates. A public key certificate is analogous to a passport, in that it proves your identity and is authorized by a trusted third party known as a Certification Authority or CA.

RAID: (Redundant Array of Inexpensive Disks). RAID technology seeks to overcome the unreliability and performance bottlenecks inherent in mechanical hard disk drives. RAID has been defined at six levels, numbered 0 through 5 and defined as follows:

0 No redundancy or data protection at all, but multiple hard drives may be spanned to create the appearance of a single logical drive.

1 Provides redundancy only, through mirroring, where a second drive is a duplicate of the first.

2 Unused, with little practical application.

3 Provides byte striping of data across multiple disk drives with parity bits for each byte stored on a separate dedicated drive; through small reads/writes, it provides performance to the maximum throughput of the parity drive.

4 Provides block or sector striping across multiple disk drives with parity blocks written to a separate dedicated drive; through small reads/writes, it provides performance to the maximum throughput of the parity drive.

5 Provides block or sector striping across multiple disk drives, but integrates the parity blocks with the data blocks, thus providing increased write performance.

Reference Architecture: An architectural description for a family of applications that describes functional components, connections, protocols, and control. A reference architecture generally consists of a partially-specified system composed of generic or abstract components, that are replaced by real components when the architecture is instantiated for an actual system (Lockheed-Martin 1997b).

Request for Comment: Typically issued by the IETF as draft documents describing a proposed capability or feature for the Internet. However, many RFCs have taken on the status of standards.

RFC: Request for Comment

SBA: Standards-Based Architecture (see the TAFIM, Volume 4, DISA 1997)

Scalability: The ability to use the same applications and systems on all classes of computers from personal computers to supercomputers.

Secure HTTP (S-HTTP): An extension to the HTTP protocol to support sending data securely over the World Wide Web.

Secure Socket Layer (SSL): A protocol developed by Netscape for transmitting private documents via the Internet. SSL works by using a private key to encrypt data that's transferred over the SSL connection.

Secured-access Transparency: Users have to be able to access any server resource from anywhere, including hotel rooms, offices, homes, and cellular phones, using a single log-on.

Security Policy: A set of rules which define and constrain the types of security-relevant activities of entities (ECMA 1989).

SEE: Software Engineering Environment

Server: A computer, or a software package, that provides a specific kind of service to client software running on other computers. The term can refer to a particular piece of software, such as a WWW server, or to the machine on which the software is running. A single server machine could have several different server software packages running on it, thus providing many different servers to clients on the network (Internet Literary Consultants, http://www.matisse.net/files/glossary.html).

Service: A function specific set of sharable code that has the following characteristics: sharable, consistently maintained, available throughout the enterprise, accessible through a published API, event-driven, and remotely accessible. Services distinguish themselves from other applications by publishing what they do, how to invoke them, and what is expected in return.

SKIP: Simple Key Management for IP protocols

SME: Subject Matter Expert

Software Architecture: An abstract system specification consisting primarily of functional components described in terms of their behaviors and interfaces and component-component interconnections. The interconnections provide means by which components interact. Architectures are usually accompanied by a *rationale* that documents and justifies constraints on component and interconnections or explains assumptions about the technologies which will be available for implementing applications consistent with the architecture.

Software Reuse: Existing software artifacts are reused when building a new software system. The artifacts include source code, design structures and notes, system architecture, test cases, documentation, life cycle processes, and system specifications.

System: A set of different elements so connected or related as to perform a unique function not performable by the elements alone (Rechtin 1991).

Systems Engineering: A management technology that controls the total system life-cycle process, which involves and which results in the definition, development, and deployment of a system that is of high quality, trustworthy, and cost-effective in meeting user needs (Sage 1992).

TAFIM: Technical Architecture Framework for Information Management, a DOD architectural framework similar to the TISAF (DISA 1997).

TCP/IP: Transmission Control Protocol/Internet Protocol. A standard set of network protocols developed by the U.S. Department of Defense and accepted as a standard network protocol worldwide, but especially for UNIX Ethernet systems.

Technical Architecture: Technical architectures are a minimal set of rules governing the arrangement, interaction, and interdependence of the parts or elements whose purpose is to ensure that a conformant systems satisfies a specified set of requirements. (This term is sometimes used in lieu of infrastructure.)

Technology: Tools or tool systems by which we transform parts of our environment and extend our human capabilities (Tornatzky and Fleischer 1990).

TISAF: Treasury Information System Architecture Framework

TP: Transaction Processing

Transparency: Techniques that help keep an application from having to know too much about its environment to function.

TRM: Technical Reference Model (of the TISAF). Described in Chapter 4.

UNIX: A family of operating systems known for their relative hardware independence and portable applications interface.

UPS: Universal Power Supply

VPN: Virtual Private Network

WAN: Wide Area Network

WBS: Work Breakdown Structure

Workflow: A way of implementing business processes that involves step-by-step progression of a task involving a number of users.

WWW: World Wide Web

APPENDIX B
REFERENCES

This appendix provides the detailed description of the references used in the preparation of this document. All URLs are current as of the date of publication of this document.

Abowd, G., L. Bass, P. Clements, R. Kazman, L. Northrop, and A. M. Zaremski. 1997. Recommended Best Industrial Practice for Software Architecture Evaluation Technical Report CMU/SEI-96-TR-025, ESC-TR-96-025, January 1997.

Barbacci, M., M.H. Klein, and C.S. Weinstock. 1997. Principles for Evaluating Quality Attributes in Software Architectures. Software Engineering Institute, Carnegie-Mellon University, Pittsburgh, PA, CMU/SEI-96-TR-036.

Booch, G. 1994. Object-Oriented Analysis and Design With Applications, Benjamin/Cummings, San Francisco, CA.

Brodie, M.L. and M. Stonebraker. 1993. DARWIN: On the Incremental Migration of Legacy Information Systems, TR-0222-10-92-165, College of Engineering, UC Berkeley, Berkeley, CA.

Bull, J. and O. Rees. 1994. A Framework for Federating Secure Systems. ANSA, APM.1006.01, Cambridge, UK.

Bull, J. and A. Herbert. 1994. Security Services. ANSA, APM.1007.01, Cambridge, UK.

Chapel, V., R.C. Aronica, and M. Alessi. 1996. Understanding Systems Architecture. The Technical Resources Consortium (http://www.trcinc.com/publications/newsletter/OS1.1/OSarch.html).

Clark, D., Pasquale, J. et al. 1996. Strategic Directions in Networks and Telecommunications. ACM Computing Surveys, 28(4):679-690.

Clements, P.C. 1995. Understanding Architectural Influences and Decisions in Large-System Projects, First International Workshop on Architectures for Software Systems, Seattle, WA.

Clements, P.C. and L.M. Northrup. 1996. Software Architecture: An Executive Overview. Software Engineering Institute, Carnegie-Mellon University, Pittsburgh, PA, CMU/SEI-96-TR-003.

Abowd, G., L. Bass, P. Clements, R. Kazman, L. Northrop, and A. M. Zaremski. 1997. Recommended Best Industrial Practice for Software Architecture Evaluation Technical Report CMU/SEI-96-TR-025, ESC-TR-96-025, January 1997.

Davis, A. 1993. Software Requirements: Objects, Functions, & States, Prentice Hall, Englewood Cliffs, NJ.

Defense Information Systems Agency (DISA). 1996. Configuration Management Software And Documentation Delivery Requirements, Version 1.0, Sterling, VA.

Defense Information Systems Agency (DISA). 1996. Defense Information Infrastructure (DII) Common Operating Environment (COE), Version 3.0, Baseline Specifications. Reston, VA.

Defense Information Systems Agency (DISA). 1997. Technical Architecture Framework for Information Management (TAFIM): Volume 4, DoD Standards-Based Architecture (SBA) Planning Guide, Version 3.0, Sterling, VA.

ECMA. 1989. "Security in Open Systems—Data Elements and Service Definitions". STANDARD ECMA-138, European Computer Manufacturers Association, 114 Rue du Rhône, 1204 Geneva.

Emery, D.E., R.F. Hilliard III, and T.B. Rice. 1996. Experiences Applying a Practical Architectural Method. MITRE, Bedford, MA.

ETC Electronic Trading Concepts Pty Ltd. 1996. User Authentication Issues in Electronic Services Delivery: Final Report, Canberra, Australia.

Fawcett, J.W. and R.W. Jepsen. 1996. Using the Common Gateway Interface. UNIX Review, 14(5):39-46.

Gacek, C., A. Abd-Allah, B. Clark, and B. Boehm. 1995. On the Definition of Software Architecture. Proceedings, ICSE 17 Software Architecture Workshop.

Gamma, E., R. Helm, R. Johnson, and J. Vlissides. 1995. *Design Patterns*. Addison-Wesley, Reading, MA.

Garlan, D., R. Monroe, and D. Wile. 1997. ACME: An Architecture Description Language. European Software Engineering Conference 97.

Garlan, D., Ed. 1995. Proceedings of the First International Workshop on Architecture for Software Systems, Seattle, WA. Published as CMU-CS-TR-95-151.

Garlan, D. and M. Shaw. 1994. An Introduction to Software Architecture. School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, CMU-CS-94-166 (http://www.cmu.cs.edu).

Gartner Group. 1991. Open Systems: Cutting Through the Confusion.

Halfhill, T.R. 1997. Cheaper Computing. Byte, 22(4):66-80.

Hayes-Roth, F. 1994. Architecture-Based Acquisition and Development of Software: Guidelines and Recommendations form the ARPA Domain-Specific Software Architecture (DSSA) Program. Version 1.01.

Herbert, A. 1994. Distributing Objects. ANSA APM 1009.01, Cambridge, UK.

Herbert, A. 1994. The Challenge of ODP. ANSA APM1016.01, Cambridge, UK.

Hewlett-Packard Co. 1996. HP Open Enterprise Computing: Executive Perspective, Cupertino, CA.

Holbrook, P. and J. Reynolds, Eds. 1991. Site Security Handbook (RFC1244). Site Security Policy Handbook Working Group (SSPHWG), IETF.

Hurwicz, M. 1997. Faster, Smarter Nets. Byte, 22(4):83-88.

Iggulden, D., O. Rees, and R. van der Linden. 1994. Architecture and Frameworks. APM Ltd., APM.1017.02.

Kazman, R. and L. Bass. 1994. Toward Deriving Software Architectures From Quality Attributes, Technical Report, CMU/SEI-94-TR-10.

King, W.R. (1995, Winter). “Creating a Strategic Capabilities Architecture,” Information Systems Management, 12(1):67-69.

King, W.R. (1978, March). Strategic Planning for Management Information Systems. MIS Quarterly, 27-37.

Lockheed-Martin. 1997. Software Architecture Technology Guide (http://www-ast.tds-gn.lmco.com/arch/arch-002.html).

Lockheed-Martin. 1997. Software Architecture Technology Guide (http://www-ast.tds-gn.lmco.com/arch/arch-003.html).

Loukides, M. 1990. System Performance Tuning. O’Reilly & Associates, Sebastopol, CA.

Malhotra, Y. 1996. Enterprise Architecture: An Overview. University of Pittsburgh, Katz School of Business, Pittsburgh, PA (http://www.brint.com/papers/enterarch.htm).

Might, R.J. 1994. Domain Models: What are They? How are they used? Institute for Public Policy, George Mason University, Fairfax, VA.

Monroe, R.T.; Kompanek, A.; Melton, R. and D. Garlan. 1997. Architectural Styles, Design Patterns, and Objects. IEEE Software, 17(1):43-52.

Montgomery, J. 1997. Distributing Components, Byte, 22(4):93-98.

Mowbray, T.J. and R. Zahavi. 1995. The Essential CORBA: Systems Integration Using Distributed Objects.

NCSC. 1987. “Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria,” NCSC-TG-005 National Computer Security Center.

Orfali, R., D. Harkey and J. Edwards. 1996. The Essential Client/Server Survival Guide, Second Edition.

OTA. 1995. Issue Update on Information Security and Privacy in Network Environments. Office of Technology Assessment, Washington, DC.

Rechtin, E. 1991. *Systems Architecting: Creating and Building Complex Systems*. Prentice Hall, Englewood Cliffs, NJ.

Rechtin, E. and M.W. Maier. 1997. *The Art of Systems Architecting*. CRC Press, Boca Raton, FL.

Riehle, D. 1997. A Role-Based Design Pattern Catalog of Atomic and Composite Patterns Structured by Pattern Purpose. Ubilab TR 97-1-1, Union Bank of Switzerland.

Rogers, G. 1982. “Information Exchange and Technological Innovation.” In The Transfer and Utilization of Technical Knowledge. D. Sahal, ed. Lexington, Mass: Lexington Books, pp. 105–123.

Sage, A.P. 1992. *Systems Engineering*. John Wiley & Sons, New York, NY.

Sandhu, R., E. J. Coyne, H. L. Feinstein, and C. E. Youman. 1996. Role-Based Access Control Models. IEEE Computer, 29(2):38-49.

Sharble, R.C.; Cohen, S.S. 1993. The Object-Oriented Brewery: A Comparison of Two Object-Oriented Development Methods. Software Engineering Notes, 18(2), pp. 60–73.

Schmidt, D.C. 1994. A Family of Design Patterns for Application-Level Gateways. Dept. of Computer Science, Washington University, St. Louis, MO (http://www.cs.wustl.edu/~schmidt/).

Shaw, M. et al. 1995. Abstractions for Software Architecture and Tools to Support Them. Computer Science Dept., Carnegie-Mellon University, Pittsburgh, PA.

Simmons, W. 1997. Rethinking Systems Management. Information Week, March 10, pp. 49–60.

Spewak, S. 1993. *Enterprise Architecture Planning: Developing a Blueprint for Data, Applications, and Technology*. QED Publishing Group, Boston, MA.

The Information Store. 1997. Web-Book on Information Architecture (http://www.istore.com/webbook/ia/index.html).

Tornatzky, Louis G., and Mitchell Fleischer. 1990. *The Process of Technological Innovation*. Lexington, Mass: Lexington Books.

Tracz, W. and L. Coglianese. 1994. Domain-Specific Software Architecture Engineering Process Guidelines, ADAGE-IBM-92-02B, Loral Federal Systems Division, Oswego, NY.

U.S. Customs Service. 1996. Automated Information Systems (AIS) Security Policy, CIS HB 1400-04, Washington, DC (http://www.customs.ustreas.gov/imp-exp/about/ais-doc.htm).

U.S. Department of Energy. 1997. Information Architecture Definition and Documentation Methodology, Washington, DC.

U.S. Department of Energy. 1997. Information Architecture Definition and Documentation Toolkit, Washington, DC.

U.S. Department of Energy. 1996. Information Architecture: Volume III, Guidance. Washington, DC.

U.S. Department of the Treasury. Treasury Security Manual, TD P 71-10, Washington, DC.

Wong, W. 1996. Create a Tough Firewall. NetworkVAR, Miller Freeman, San Francisco, CA, 4:10, pp. 34–39.

Wood, C.C. 1997. Information Security Policy Made Easy. Baseline Software, Inc., Sausalito, CA.

Zielinski, K., G. Czajkowksi, and A. Uszok. 1994. Interoperability in Open Distributed Processing. UMM Institute of Computer Science.

APPENDIX C
ARCHITECTURAL DOCUMENT TEMPLATES

This appendix presents a set of templates for the architectural documents that were identified in section 3 as either deliverables or work products. Deliverables are documents that are formal products of tasks or activities of the architecture development process. Work products are interim or temporary products produced in support of or as early versions of deliverables.

C.1 ARCHITECTURAL DEVELOPMENT PROJECT CHARTER

Architecture Development Project Charter Template

_____

Project Leader Date

_____

Senior Executive Date

Table of Contents Page Number

1 INTRODUCTION

This document establishes the boundaries of the project and the commitment of various organizations responsible for the planning and execution of the project. Boundaries need to be defined so that the project team understands which organizational and system elements are included and excluded from the architecture. Commitment needs to be documented so that participants can see that they have the support from those in the organization who have the authority and resources for development and implementation of the architecture.

2 PURPOSE

The charter represents an agreement between the senior level executive sponsoring the project and the project leader. At the enterprise level, the senior executive is the system information resource management office (SIRMO) or chief information officer (CIO). The details in this charter provide the organizational and administrative framework within which the architecture will be developed. The principle participants must be able to understand, from this document, why the architecture is needed, what it is expected to accomplish and how it will benefit the organization.

3 BACKGROUND

This section summarizes the business/computing activities that explain the background and provide the groundwork for the development of the project. The background includes the motivation for development of the architecture as well as any relevant business, project and organizational history. This provides a setting which puts the architectural development effort in context and provides a setting for the development effort.

4 SCOPE

This section defines the organizational elements, missions areas and/or functional areas that are to be addressed. Scope identifies the breadth and depth, in terms of the applications, the information systems, the organization and work processes, the architecture is to address. All of these, where relevant, should be described in this section to indicate limits and applicability of the architecture.

5 OBJECTIVES

This section describes the objectives and may include a preliminary vision statement of the future enterprise and the desired computing environment. The objectives may be operational, such as to reduce the time to respond to public inquiries; technical, such as to increase the capacity of archive storage media or administrative, or administrative such as to increase access to specialized reports or a combination of these and others. They should be enumerated and any additional explanation surrounding their generation or purpose should be included. It is particularly important that the objectives be clear since it forms the backbone of the charter.

6 EXPECTED OUTCOME

The expected benefits and other outcome of the project are described in this section. This may include a description of the success criteria and the anticipated level of achievement with respect

to those criteria. As much detail as possible should be included to ensure that those agreeing to this charter have a clear vision of the impact of the project on the operations of the organization. Examples of such criteria are:

- Achieve a 20% reduction in the length of time for the average public inquiry to be satisfied
- Increase the number of staff with access to the LAN by 50%
- Improve the ability of administrative staff to share resources and data by 25% over 18 months.

Glossary
Index

C.2 ADP TEAM ROSTER

Roster of ADP Team Members

Date:_____

Architecture Development Project:_____

Core Team Members

	Background: Mission/Business/Information Systems M/B/IS
1.	
2.	
3.	

Subject Matter Experts (SMEs)

	Expertise
1.	
2.	
3.	

Support Personnel

	Role
1.	
2.	
3.	

C.3 ARCHITECTURAL DEVELOPMENT PROJECT PLAN

Architecture Development Project Plan (ADPP) Template

Table of Contents Page Number

1 INTRODUCTION

This document establishes the boundaries of the architecture development project and the commitment of various organizations responsible for the planning and execution of the project. Mention should be made of the project charter for additional information on the nature of the agreement among the stakeholders and the highest levels of the management team.

1.1 PURPOSE

The purpose of the project plan is to provide a rough estimate of the level of effort for core team members and subject matter experts, the time required and the estimated cost to complete the project.

1.2 BACKGROUND

This section summarizes the business/computing activities that explain the background and provide the groundwork for the development of the project. This information should be consistent with the material presented in the project charter. Information that is particularly relevant to the architecture development effort should be expanded and references provided to back-up material, manuals and existing projects and plans.

1.3 OBJECTIVES

This section describes the objectives that need to be met and their priority if known. The objectives in this plan should be consistent with and described at a more detailed level than the objectives in the project charter. For instance, the objective to reduce the time to respond to public inquiries from the project charter might be include the following objectives at the project level:

- Incorporate open systems standards in the external query response system
- Upgrade the LAN network infrastructure
- Convert the query database query utility to a standard query language

1.4 SCOPE

This section defines boundaries of the project and any limitations that may affect the successful completion of the project. These should be consistent with the scope expressed in the project charter.

Any budgetary limitations as well as an overview of the budget for the project is also described in this section. This may be presented as an initial budget amount with an anticipated spending plan to show how the project expects to draw down funds. Any interfaces with the organization’s budgeting process should be explained.

1.5 REFERENCES

Any background documents, policies, standards or procedures to be followed on the project are listed in this section. This includes system documents, users manuals or programming manuals for the applications affected by the project's tasks.

2 RISKS

Potential risks from a management, administrative and technical view should be described (such as a tight schedule, unavailability of personnel, limited funding, etc.). If the project is sufficiently large, this section should describe the overall approach to risk management and point to the development of a separate risk management plan.

Risk Category	Risk	Explanation
Management	Project reviews are scheduled every month.	A lot of time is required to prepare a presentation; it's possible this will interfere with effective management of the project.
Administrative	There is no agreed upon process for delivery of products.	The project needs to know how may copies, who they go to and the turnaround time for comments to be incorporated if delivery is to go smoothly.
Technical	The need for skilled Java programmers will occur before the training can be scheduled.	If the current schedule holds, the Java programming task will be started before the training can be taken.

Steps to mitigate these risks and manage them is described in the project's risk management plan.

3 ROLES AND RESPONSIBILITIES

This section describes the organization structure of the project. In particular, the assignments for the individual ADP team members (see the Roster of ADP Team Members) must be defined.

Core Team Members	Position	Background: Mission/Business/ Information Systems
Jane Doe	Task Leader	Project management; budget development, risk assessment
Harvey Ball	QA Team Leader	
Subject Matter Experts (SMEs)	**Position**	**Expertise**
Stanley Kowalski	Lead Consultant	Electronic Commerce
Support Personnel	**Role**	**Skill**
Ann Rice	Technical Editor	Technical editing; copywriting

A project organization chart should also be developed and included in this plan.

4 WORK BREAKDOWN STRUCTURE

This section includes the work breakdown structure (WBS) for the project. A WBS is a hierarchical decomposition of the task into separate, logical and manageable tasks and subtasks. In a typical WBS, the highest level refers to the entire project, the second level is composed of major subprojects and the third and lower levels define deliverables. Each element in the hierarchy is mapped to a task. At the lowest level, schedules can be generated and costs assessed for roll-up to a higher reporting levels. In addition to the graphical depiction of the WBS, each task is described in the following sections of the document. This includes a description of how the available resources, people and tools are allocated to the different activities of the architecture development project plan (ADPP). Tasks that should be included in this section are project reviews and internal team reviews. The following figures are examples of a WBS. The first depicts the first two levels and the second shows the detail for element 01.03.

(INSERT WBS FIGURES HERE)

4.1 TASK 01.01.01

This narrative describes the activities that are required to accomplish this task. The information is available for input to a scheduling program. It is at the activity level that cost information will most likely be collected and aggregated for reporting at a higher level. This section is repeated as often as necessary to include all relevant tasks in the WBS.

5 DELIVERABLES

The products, documentation and other items to be produced by the project and made available to outside groups are listed in this section. Dates may be included if they are not to be part of the schedule. Each product is associated with an element of the WBS.

WBS Element	Work Product	Due Date
01.01.05	Training Schedule	1 month after start of project
01.02.06	Draft Baseline Architecture	4 months after start of project

Deliverable	Due Date
Feasibility Study	60 days after start of project
Baseline Architecture	6 months after start of project
Target Architecture	18 months after start of project

6 RESOURCES

This section describes all the resources required to complete the project. This includes personnel, hardware, software, tools, facilities and training. Each resource category should address the amount of the resource required, when it is needed, when it is no longer needed and its intended use. An example follows:

Resource	Amount	Date Needed	Date No Longer Needed	Intended Use

PC laptops	12	Start of project	End of project	Requirements definition; Prototype demonstration
Prototyping Software	5 seat license	2 months after start of project	End of project	Develop proof of concept and prototypes of possible architectures
EDI Consultant	1	1 month after start of project	2 months prior to end of project	Advise on structure of electronic commerce segment of the architecture

7 SCHEDULE

The schedule identifies the timeframe for each task and the dates when deliverables are due. It should include lead time for product and documentation preparation. If a scheduling tool is used, this section should contain the output from that product or refer to an appendix where the project schedule is documented. The information should be consistent with any time frames defined in the project charter.

8 DOCUMENTATION AND PUBLISHING STANDARDS

This section describes the standards to be followed in preparing the documentation for the project. It also describes the standards to be followed in publishing deliverables and other output from the project. The standards are listed according to whether they are mandatory or optional.

Standard	**Purpose**
Mandatory	
Treasury Documentation Standard	Format for deliverable documentation
Optional	
Frontpage 97 (Microsoft)	Publish web pages

9 COST ESTIMATE

This section should include an estimate of the level of effort for core team members and subject matter experts, time required and estimated cost. If a cost estimating system is used, this section can refer to an APPENDIX which would contain the output from the estimating system.

10 CRITICAL SUCCESS FACTORS

This section may be included to describe the set of functions and features that will make each stakeholder feel their needs have been met by the project. These must be consistent with the critical success factors from the project charter. In addition, this section should describe which elements of the architecture are expected to support the individual critical success factors, insofar as this is known. For instance decreasing response times for external queries might be associated with a specific component of the infrastructure architecture within the target architecture.

Glossary
Index

C.4 ADP AUTHORIZATION MEMORANDUM

Interoffice Memo
DATE: 07/17/97
To:
CC:
From: Authorized User
RE: Architecture Development Project Plan
The XXXXXX Architecture Development Project is hereby authorized.

Senior Management/Senior Executive

AU/AU

C.5 ENTERPRISE BUSINESS VIEW

Enterprise Business View Template

List of Figures

List of Tables

1 INTRODUCTION

This document contains a description of the enterprise's business operations and its operating environment. The objective is to provide an overview of the roles and responsibilities of the organization, the functions it carries out, the products and services produced and the external entities with which it interfaces.

1.1 PURPOSE

The purpose of the enterprise business view is to identify and specify the elements comprising the business' operations, its locations, functions and relationship to external organizations. This description may incorporate changes which occurred due to an iteration of the architecture development process or the corresponding system development process, such as a reorganization, the addition of new missions or goals, or revised business processes and relocated business operations. For example, this document might describe a new organization created by a merger of the customer relations and marketing organizations.

1.2 SCOPE

This section describes boundaries of the business view in terms of the organizational units, external entities, and relationships among them. For example, the scope of the business view described in the preceding paragraph is the current operations of the newly merged customer relations and marketing organizations. This view may be presented in several layers. The degree of business view decomposition reflects the size and complexity of the organization, the breadth of the mission objectives and the perceived future needs of the ADP team.

1.3 REFERENCES

This section contains a list of documents related to the topics covered in this material. This may include detail level documents for the business operations. The formal organizational structure for the bureau or organizational unit/level may be included as well as mission statements.

1.4 ORGANIZATION OF THE DOCUMENT

Following the introduction and background, section 2 describes the organization structure. Section 3 provides a high-level description of the business view in terms of the major internal and external entities and the fundamental interfaces between them.

2 ORGANIZATIONAL STRUCTURE AND BUSINESS OBJECTIVES

The organizational structure consists of a set of business units and the personnel responsible for the operations within those units. This included organizational components that provide services and products to external customers as well as those that provide infrastructure support to other components of the enterprise.

2.1 ORGANIZATION STRUCTURE

The business organization chart is provided in figure 1. It identifies staff and their positions/titles/locations and includes annotations for other pertinent information.

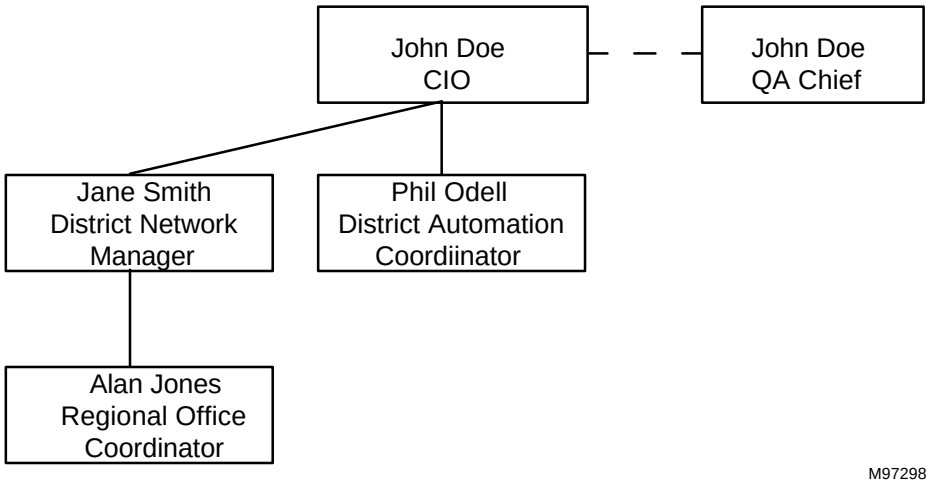

Figure 1. Enterprise Organization Chart

The following figure identifies physical business locations relevant to the enterprise and the relationship of those locations to the organizational units (who reports to whom).

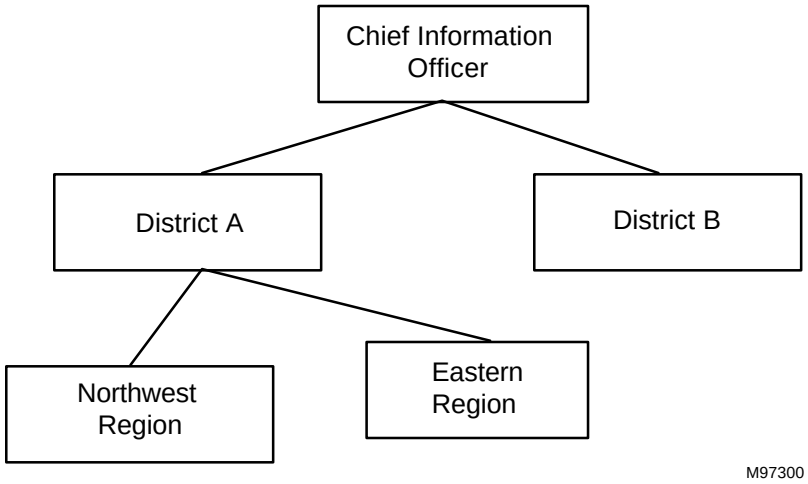

Figure 2. Relationship of Business Locations to Organizational Units

In addition to the organizational units shown in figure 1, the following chart shows the relationship of several working groups, committees and ad hoc teams to the organizational units of the enterprise.

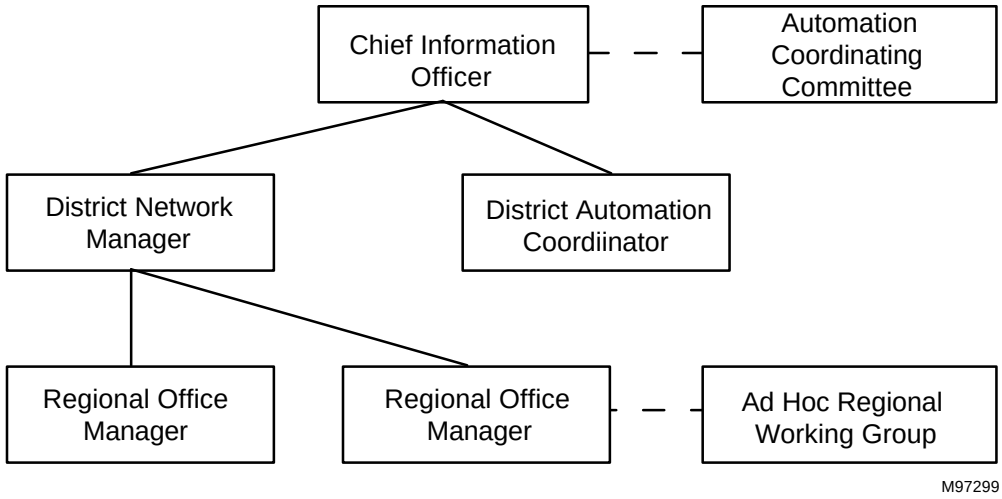

Figure 3. Identification of Oversight Committees

2.2 BUSINESS OBJECTIVES

The business should have a defined set of objectives that guides development of its strategic plan and the definition of its organizational units as well as its activities, products and services. In this example, the current enterprise business objectives include the following:

- Increase productivity of query processing.
- Ensure interoperability of office automation software.

3 THE ENTERPRISE BUSINESS VIEW

The organizational structure represents the physical structure of the organization including information such as where business operations are located and who reports to whom. The organization's mission objectives flow down from the enterprise level to business operations. In this section the organization structure is associated with specific objectives through a description of roles and responsibilities. The responsibilities of each organizational unit are defined to facilitate the association of business objectives with responsible organizational units. The following example illustrates how this can be done:

• Chief Information Officer	Responsible for the direction and composition of the hardware and software infrastructure supporting the staff activities.
• District Network Manager	Responsible for the communications infrastructure among the regional offices.
• Regional Office Manager	Responsible for the achievement of production goals for each regional office. The regional office manager is also responsible for communications to the district manager.

Once the responsibilities for each role are understood, objectives can be associated with the appropriate organizational unit as illustrated in the following table.

Table 1. Relationship of Business Objectives to Organizational Units

Objective	Organizational Unit
Increase productivity of query processing	Regional Office
Ensure interoperability of office automation software	Chief Information Officer

3.1 LIST OF PRODUCTS AND SERVICES

Part of the business view is understanding what the products and services are and how they relate to the organizational units. This understanding is achieved through a step-by-step process of identifying each element and associating them with each other. First the products and services are identified as shown in this example:

Products: Annual Customer Profile
Product Brochures

Services: Customer Query Support
Market Analyses

The next step is to identify the customers who receive those products and services.

3.2 CROSS REFERENCE OF CUSTOMERS AND PRODUCTS/SERVICES

The following table shows the relationship of the organization's products and services to recipients of those products and services. It also show whether those customers are internal or external entities.

Table 2. Relationship of Products/Services to Customers

Product/Service	Customer	External/Internal
Annual Customer Profile	Vice President of Operations	Internal
Product Brochures	Preferred Customers	External
Customer Query Support	General Public	External
Market Analyses	Vice President of Marketing	Internal

3.3 CRITICAL EXTERNAL BUSINESS PRODUCTS AND SERVICES

The business enterprise often relies on products and services from external entities to conduct the business. These are identified in this section. The following table is an example of how this can be illustrated.

Table 3. Relationship of Products and Services to Providers

Product/Service	Provider
Regional Census Report	State Census Bureau
Custom brochure stock	EZ Print Services
Mailing lists	Capital Mailing Lists
Color Print Consulting	EZ Print Services

The next step is to associate these external entities and products or services with the business unit that uses them.

3.4 RELATIONSHIP OF CUSTOMER AND PROVIDERS TO ORGANIZATIONAL UNITS

The following matrix relates the customer and providers to the organizational units of the business through the services and products that are provided.

Table 4. Relationship of Customers/Providers to Organizational Units

Provider	Product/Service	Organization Unit
State Census Bureau	Regional Census Report	Marketing
EZ Print Services	Custom brochure stock	Public Relations
Capital Mailing Lists	Mailing lists	Marketing
EZ Print Services	Color Print Consulting	Customer Relations
Customer		
Vice President of Operations	Annual Customer Profile	Corporate Operations
Preferred Customers	Product Brochures	Public Relations
General Public	Customer Query Support	Customer Relations
Vice President of Marketing	Market Analyses	Marketing

3.5 ENTERPRISE-WIDE BUSINESS VIEW

This section contains an enterprise-wide diagram of the business operations showing how they relate to the customers and providers of businesses and services. A high-level view of the business is presented below as an example. In this diagram, the major organizational units and their external customers or suppliers are shown. In this example, the major organizational units would include a reference to a lower level diagram in which that units roles, responsibilities, products, services, customers and providers are described. Those references may be pointers to tables similar to the ones used as examples in earlier sections of this template. In this way, the internal structure of the separate business operations are described and brought together in a single view of the business enterprise.

INSERT DIAGRAM OF BUSINESS ENTERPRISE.

Acronyms
Glossary

C.6 BASELINE WORK ARCHITECTURE

Baseline Work Architecture Template

Title Page No.

List of Figures

Title Page No.

List of Tables

Title Page No.

1 INTRODUCTION

This document is a high-level description of the current business functions of the enterprise. Business functions are any set of actions that are performed in the conduct of the business. Business entities are objects representing information that can be manipulated by the functions or performers or providers of information. This step is internally focused on the enterprise's business processes.

2 WORK ENTITIES

Work entities are objects that perform some action and/or represent a set of information that can be manipulated by the business components and are within the boundaries of the architecture. The objective is to document the relationships among work entities and the organization structure. Graphics are an important part of this section. An enterprise-wide diagram or chart should be included. It should be consistent with any enterprise-wide diagram included with the enterprise business view document. The following is a portion of an enterprise-wide work entities diagram. Detailed level descriptions of the work entities can be provided in lower level diagrams and aggregated to the higher level enterprise wide diagram.

(INSERT DIAGRAM:
SAMPLE PORTION OF AN ENTERPRISE WIDE WORK ENTITIES DIAGRAM)

Where a diagramming notation has not been selected, ones associated with functional decomposition and object-oriented methodologies provide rich notations for this view. The information to be included is described in the following paragraphs with illustrations of how each type of information might be presented.

2.1 WORK ENTITY: REPORT EXTERNAL QUERIES)

The first element in the baseline work architecture are the work entities. Each entity that can be manipulated by the business units is described separately. The following example illustrates one way to describe the entity. It is organized in a table format with the name of the entity and attributes which provide additional information about it.

Table 1. Example of Work Entity Data

Name	Report external queries
Location	Regional and district offices
Subfunctions	Log call Update customer call record Track call record
Frequency	Hourly
Inputs	Customer data
Outputs	Call record
Responsible organization	Customer services
User	Quality assurance
Actions	Review for adherence to standard

When all the work entities are described, they should be associated with the organizational unit in the business enterprise that is responsible for implementing them. This is done through a matrix as illustrated in the following table. This matrix summarizes the work entities and associates them with the organizational unit and staff responsible for their execution. Following this table is an example of a diagram showing how the individual entities relate to each other at the lowest level of definition. They should aggregate to the enterprise-wide work entities diagram illustrated in section 2.

Table 2. Work Entities Mapped to Organizational Units and Staff

Work Entities	Organizational Unit	Responsible Staff
Report source and number of external queries	Regional Office	Help Desk Officer
Manage the query database	Data Base Administration	Database Administrator

Figure 1 is an example of the relationship between two entities using entity relationship notation.

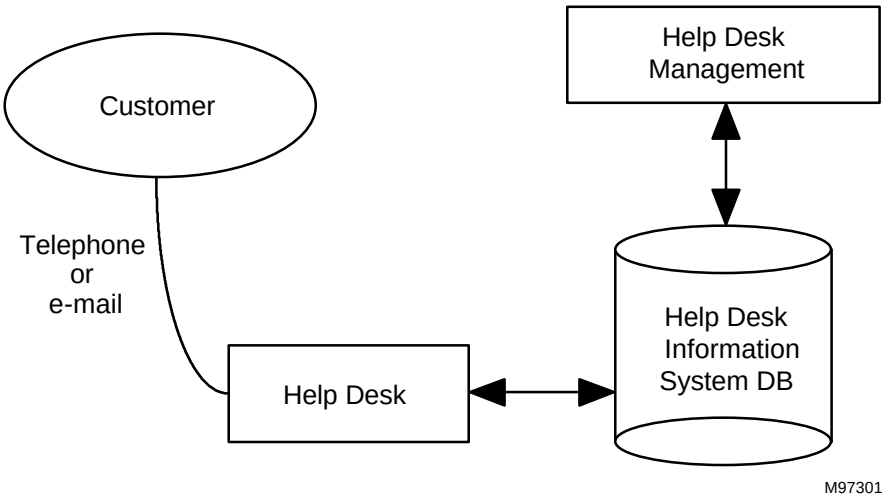

Figure 1. Communication Flow for Query Support

3 EXTERNAL WORK ENTITIES

Similarly to the description in section 2.1, entities external to the business enterprise need to be defined. These external functions and/or entities provide input, receive output or otherwise interact with internal business functions and entities. They are outside the boundaries of the enterprise but interact with it. The interfaces define the boundaries of the baseline work architecture. Defining the external entities helps to identify the boundaries of the baseline architecture.

3.1 EXTERNAL WORK ENTITY: REGIONAL CENSUS DATA (REPEAT AS NECESSARY)

External work entities are those items that provide sets of information that can be manipulated by the internal work entities of the business enterprise. They can be described in a similar manner to the internal entities as illustrated in the following table.

Table 3. Example of External Work Entity

Name	Regional Census Data; provides citizen demographic information to the enterprise
Location	State census office
Frequency	Annual
Inputs	Annual census survey data
Outputs	Annual census survey
Contents (if used by different internal business components)	N/A
Communication Medium	U.S. mail

3.2 INTERACTIONS BETWEEN INTERNAL AND EXTERNAL WORK ENTITIES

Interactions are information flows that occur between internal and external functions. In the following example, the different information entities are identified along with the relationships between the entities that participate in the interaction. An entity relationship diagram is used to illustrate this concept.

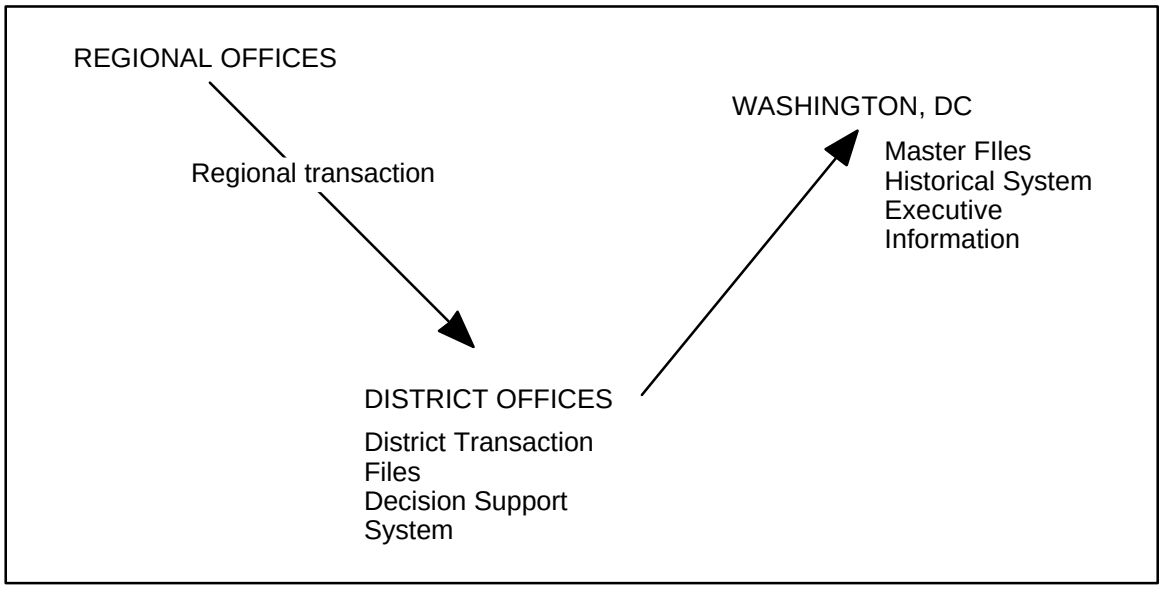

Figure 2. Interactions Among Business Entities

Glossary

C.7 BASELINE FUNCTIONAL ARCHITECTURE

Baseline Functional Architecture Template

Title Page No.

List of Tables

Title Page No.

1 INTRODUCTION

The baseline functional architecture describes the current business applications that support the current business functions, and identifies and defines the data and information entities. This document documents the results of that analysis. This effort is a high-level description, not a low level specification. It should be correlated with the development of the information architecture.

2 BUSINESS APPLICATIONS

This section documents work done to collect and organize information about the current business applications. The following information is provided for each application:

- Application name and description
- Files and databases used by the application
- Short name, abbreviation, or code for the application
- Personnel responsible for maintaining the application
- Primary users of the application (by business operation)
- Business functions supported by the application
- Resources allocated to support and maintain this application

The following table provides an example of this information relevant to a specific application.

Table 1. Example Business Application Information

Application name and description	Query Tracking System
Files and databases used	Query Master File, Query Archive Files
Short name for the application	QTS
Responsible maintenance personnel	Customer Support Department
Primary users	Customer Interface Staff
Business functions supported	Customer relations, sales, inventory
Resources allocated this application	2 maintenance staff; 2 gigabytes of on-line storage

3 BUSINESS APPLICATION RELATIONSHIPS

This section documents which applications support which business functions. This is necessary to ensure that all applications support functions important to the business and which functions are supported by applications. This is shown in the matrix below. Business functions not supported by any current application should be highlighted. They should be considered in the design of the target architecture as an opportunity for enhancement of the business application suite.

Table 2. Example Application to Business Function Matrix

Business Application	Customer Inquiry Function	Quality Assurance Function	Management Evaluation Function
Query Tracking System	X	X	
Performance Reporting System		X	X
Review Scheduling System			X

It is also important to know which data is used by the applications and how the data that the enterprise has stored is used. This information can be described in a table or matrix format. As an example, the following matrix shows which of our example applications use which files and databases.

Table 3. Example Business Application to Database Matrix

Business Application	Query Database	Performance Measurement Database
Query Tracking System	X	
Performance Reporting System		X
Review Scheduling System		

In the same way, it is important to show how the applications interact with each other. A table or matrix is also a convenient and effective method to use to show this information. The following table is an example of how this might be done. It shows which applications from our examples use information from other applications.

Table 4. Example Business Application Cross-Reference

Business Applications	Query Tracking	Performance Reporting	Review Scheduling
Query Tracking System		X	
Performance Reporting System			X
Review Scheduling System		X	

In addition, this section should describe the role of each of the business units in using the applications in general terms such as whether they read, update or modify data. This section provides definitions to express what the business application does or should allow business units to do. This entails reviewing existing definitions and identifying redundant capabilities/applications. The following table provides an example of how this could be illustrated. In this example, a CRUD diagram could be developed for this section. The following diagram shows which organizational unit is responsible for creating [C], reading [R], updating [U] and deleting [D] the entity defined in the example.

Table 5. Organizational CRUD Diagram for Report Source and Number of External Queries

Organizational Unit	Help Desk Staff	Help Desk Management
Business Entity		
Query	C	
Log Form		C, U
Log Record	C, R, U	D

4 FUNCTIONAL AREAS

A functional area is a set of closely related business functions. They should be defined for the enterprise and documented in this section. A diagram showing which functions are performed by which business units should be included with a brief narrative and glossary of the terms used in the description. It should be decomposed to no more than three levels.

(INSERT SAMPLE DIAGRAM ON THE SUPPORT FUNCTIONAL AREA)

4.1 FUNCTIONAL AREA: SUPPORT

The example provided here describes each functional areas in terms of its products and services, customers and providers. The following narrative shows the functional area, including the following information.

- Products and services provided by each functional area
  - ◊ Log form
  - ◊ Query summary reports
- Responses to queries
- A list of the customers (internal and external) receiving the products and services
  - ◊ Users of the Help Desk
  - ◊ External Customers
  - ◊ Managers
- A list of the providers of products and services required by the functional area(s)
  - ◊ N/A
- A matrix showing how the customer and providers are related to the functional area(s) through the products and services

Customers **Products/Services**

Users of the help desk	Query responses
Managers	Management reports

5 ANALYSIS

This section includes a description of how the services and products provided by the functional area(s) can be improved with respect to cost or value. Each recommended enhancement should include a description of how the business will benefit from its implementation. In addition, an estimate of the cost and benefit of each improvement should be included in this section. If a cost/benefit analysis tool is used, this section should include the report generated by the system with an explanation of how the figures were derived.

C.8 BASELINE INFORMATION ARCHITECTURE

Baseline Information Architecture Template

Title Page No.

List of Figures

Title Page No.

List of Tables

1 INTRODUCTION

The purpose of this document is to (1) document the major kinds of data and information needed by the organization and their interrelationships, and (2) formulate a corporate data model and data dictionary.

2 THE CORPORATE DATA MODEL AND DATA DICTIONARY

The corporate data model (CDM) and data dictionary (DD) provide a list of current data entities (people, places, things, and events) used by the enterprise and additional details about those entities. The following resources might be used to develop this list for each function:

- Preliminary data model/data dictionary if available
- Information architecture catalog, data stores and existing applications systems
- Backups and products from other architectures and activities

The data entities are documented through the use of entity-relationship diagrams as illustrated below.

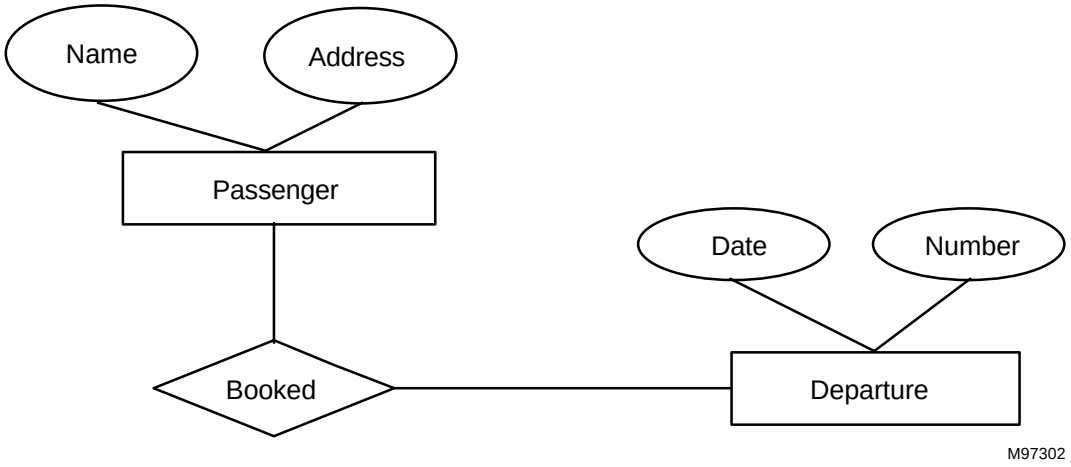

Figure 1. Example Entity-Relationship Diagram

The breadth and depth of this list (e.g., the amount of detail provided) depends on the amount of data and information entities, the size and complexity of the enterprise, and the degree of detail required to begin the next iteration of the ADP.

2.1 CRUD MATRIXES

This section documents detailed CRUD (Create-Read-Update-Delete) matrixes that relate the entities to business functions. External data should be excluded from these matrixes unless the data is used internally within the enterprise. There is no need to maintain information on data not considered important to the enterprise's functions.

Table 1. Example CRUD Diagram

Business Function	Log Query	Review Log Records
Business Entity		
Query	C	
Log Form		C, U
Log Record	C, R, U	D

Other information that should be included in this section is as follows:

- Relationship of entities to existing inputs, outputs, files, databases and information sources. This provides traceability among the various elements of the information architecture.

- Entities used exclusively in manual processes. This information can be used in the analysis to identify areas to improve or enhance for the target architecture.

- Redundant entities. These should be analyzed and eliminated in the development of the target architecture.

Additional information that may be provided in this section includes sizing and volume information, backup and recovery requirements, and integrity and security constraints. Specific information for this section is contained in the categories in the following sections. This presents a complete picture of the information used by the business enterprise.

2.2 DATA

- Data dictionary
- File structures
- Programming languages
- Storage (capacities, access methods, media, techniques)
- Dissemination/distribution
- Backup and recovery
- Integrity
- Security

2.3 DATA MANAGEMENT

- Records management
- Data centers
- Data flow
- Ownership
- Migration
- Exchange/interchange
- Manipulation

- Duplication
- Replication
- Synchronization
- Standardization
- Contingency/disaster recovery/continuity of operations
- Accessibility
- Vulnerability
- Portability

C.9 BASELINE INFRASTRUCTURE

Baseline Infrastructure Template

Title Page No.

Table 1. Example Functional Area Computer Support Matrix

1 INTRODUCTION

This document contains an analysis of the current business infrastructure. It describes the current data stores, computer systems, level of computer support for the business' applications and the strengths and weaknesses of the current infrastructure.

2 INFRASTRUCTURE INVENTORY

This section contains the infrastructure inventory. This information is necessary to enable the analysts to propose enhancements and improvements to the existing architecture. Without this knowledge, the analysts will not be able to determine where improvements could be made or suggest the best architectural solutions. The following items are included as appropriate:

2.1 PLATFORMS

Graphics or schematics showing the elements in each platform suite help effectively communicate the information. It is suggested that graphics be used whenever possible in the following sections.

- Intelligent WAN systems
- Establishment-based switching systems
- Local area network systems
- Enterprise or corporate processing systems
- Divisional or departmental processing systems
- Desktop or portable intelligent workstations

2.2 APPLICATIONS

Application is used here in the generic sense of type of automated support. As can be seen from the following list, this section describes the types and levels of automated support available in the business enterprise. Each type of support should be accompanied by the specific application or business entity it automates.

- Batch processing
- Computer conferencing
- Decision support
- Document processing
- Document storage and retrieval
- Electronic mail
- Electronic publishing
- Enhanced telephony
- Expert systems
- Hypermedia processing
- Inquiry processing
- Real-time control
- Shared screen teleconferencing

- Text processing
- Transaction processing
- Video processing
- Video teleconferencing
- Voice mail

2.3 TECHNOLOGY

Technology, as used here, is concerned with the use of automated tools and techniques to enable and manage activities. Where appropriate, include graphics or schematics to illustrate the use of the technology.

- Communications management
- Conferencing management
- Database management
- Development management
- Distribution management
- Document management
- Hypermedia
- Repository
- System management
- Transaction management
- User interface services

3 FUNCTIONAL AREA SUPPORT

This section includes information on the computer systems that support the applications for each functional area. It is necessary to provide this information so that analysts can assess the need for enhancements or improvements in the target architecture. To do this, the following information should be provided for each computer system. This is illustrated in the sections below.

3.1 FUNCTIONAL AREA A (REPEAT AS REQUIRED)

Table 1. Example Functional Area Computer Support Matrix

Functional Area A	Resources Required	Rationale
Computer System A:	Specifications	This configuration is required because of the large number of graphics processed by the application.
	System software	
	Utilities	
	Programming languages	

3.1.1 Product Catalog

This section includes references or information on existing products, applications, software, databases and other components that form the computer system for the functional area.

3.1.2 Technology Improvement Projects

This section describes any projects planned for the functional area that will enhance the computer systems. The analysts responsible for the target architecture will need to be aware of this when defining the new system.

3.1.3 Technology Assessment

This section documents the level of technology currently provided to the functional area in support of its operations. This information is required so that the analysts can assess the level of effort required to enhance the functional area. If the technology is quite old, the analysts might choose to recommend implementing the target architecture in phases, or in small increments that are more easily assimilated into the other elements of the target architecture.

3.2 BUSINESS OPERATIONS MATRIXES

This section documents the relationship between the physical technology infrastructure and the business operations that own and/or use them. Matrixes are a good way to show the relationship of systems to business operations, to applications, and to physical locations.

As appropriate, includes graphical schematics of the relationships between systems that are annotated by the major information flows that occur among them.

C.10 BASELINE ARCHITECTURE

Enterprise Baseline Architecture Template

1 INTRODUCTION

This document contains a description of the bureau's current information systems and the operating environment. It consists of the collection of individual baseline architectures developed by analysts and described in the preceding templates. They are combined here into a single document. Refer to each document for details.

1.1 PURPOSE

The purpose of the baseline architecture is to identify and describe the elements comprising one or more information systems, work, functional and infrastructure architectures supporting a bureau's activities. It describes the system as it exists today and is known as the baseline architecture.

The inventory of the organization's current information systems is useful for identifying hidden assets, gaps and redundancies, managing business costs, finding out who is using what and why, and classifying, by value, the business assets related to information technology. A clear view of the existing architecture allows the organization to begin to identify opportunities for change as it refines its vision of the target architecture.

1.2 SCOPE

This section describes boundaries of the baseline architecture in terms of its relationships to other existing information systems, ongoing information system development and the timeframe within which it is expected to be valid. The scope should be consistent with the scope of the separate architectures as described in the individual documents. The intended audience for this document is senior information technology managers.

1.3 BACKGROUND

This section describes the business needs that resulted in the creation of the information system or are currently satisfied by the operation of the information system. Any history relevant to the design of the information system or its operation is described. Information may include references to other documents or projects as well as to standards, tasks or future plans for the system. This background should be consistent with the background descriptions in the other documents.

1.4 REFERENCES

This section contains a list of documents related to the topics covered in this material. This may include detail level documents for individual information systems or business operations other than those included in the references for the individual architecture reports. For example, the formal organizational structure for the bureau or organizational unit/level may be included as well as mission statements.

1.5 ORGANIZATION OF THE DOCUMENT

Following the introduction and background, Section 2 describes the enterprise business model. The next three sections are the three EISA views. Section 3 is the work architecture, Section 4 is

the functional architecture and Section 5 is the information architecture. The technology infrastructure and information system description are in Section 6. Taken as a whole, these documents form the baseline architecture document for the enterprise.

2 ENTERPRISE BUSINESS VIEW

Documented in Appendix C.5.

3 BASELINE WORK ARCHITECTURE

Documented in Appendix C.6.

4 BASELINE FUNCTIONAL ARCHITECTURE

Documented in Appendix C.7.

5 BASELINE INFORMATION ARCHITECTURE

Documented in Appendix C.8.

6 BASELINE INFRASTRUCTURE

Documented in Appendix C.9.

Glossary
Index

C.11 BUSINESS REQUIREMENTS DOCUMENT

Business Requirements Document Template

1. Introduction
1.1 Purpose
1.2 Scope
1.3 Background
2. Requirements
2.1 Functional Requirement A
2.2 Performance Requirements
2.2.1 Constraints
2.2.2 Resource
2.2.3 Hardware
2.2.4 Software
2.3 Standards
2.4 Limitations
2.5 Other Requirements
3. Impact of Requirements
4. Issues, Risks and Alternatives

1 INTRODUCTION

As systems evolve, the functions they were originally designed to support may change. When this occurs, one of the first tasks is to determine which requirements have changed and how the system needs to be modified to support them. This document is the vehicle for documenting those changes.

1.1 PURPOSE

The purpose of the business requirements document is to describe the requirements, from the enterprise's business perspective, that lead to a change from the existing baseline architecture to the target architecture. There are two primary sections to the document. The first is the description of the new requirements. The second is the description of how those requirements are met by the current baseline architecture. This is to show the deficiencies in the current systems and provide guidance to the target architecture development team.

1.2 SCOPE

This section describes the limits of the business requirements description document. It may be limited in terms of functional area, information system, or architecture. In addition, it may apply to a specific timeframe or project. The following are described in this document:

- A definition of the boundary between system and environment for each information system represented by the architecture;
- The information that flows between each information system and its environment and the semantics of that information
- The set of features offered by each information system, e.g., the business needs it will support; and,
- The behavioral and qualitative requirements to be met by each information system.

1.3 BACKGROUND

The background section includes rationale for development of the requirements document and any environmental data or descriptive material. This may include references to changed mission statements, new organizational objectives, the need to upgrade basic infrastructure or modernize existing processing.

2 REQUIREMENTS

This section defines the strategic goals, problems, and critical success factors of the organization and develops the future picture of the business architecture. These business requirements characterize how the organization wants to operate in the future.

The business requirements are classified into two categories:

Functional requirements. These are requirements that are stated using a simple input-output model. An input changes the system state which results in an output that is generally independent of time. Intuitively, these requirements might be verified by watching the system perform in all of its states and comparing its outputs against some standard of correctness. These requirements might include safety, security, fault-tolerance, reliability, and availability, which are called functional quality attributes, because they enable the system to produce correct output in each state of the system's operation.

Performance requirements. These are time-dependent requirements that can be checked by observing the behavior of the system. In these requirements, changes to the system's state are dependent on time. Nonfunctional requirements express the need to provide certain system qualities such as maintainability, openness, portability, etc.

Each business need, known as a functional or nonfunctional requirement, is described in this section. They should be described with sufficient detail so that the next level document or design can be produced from its contents. They are described through various characteristics as shown in the following example. The output of this step is this business requirements document. This document identifies only the new or modified business requirements that have emerged since the last iteration of the ADP.

2.1 FUNCTIONAL REQUIREMENT A

The requirement is described through various characteristics as listed below. These characteristics are important to enable a proper design of the target architecture. Without this information, the designers cannot determine which parts of the existing architecture should be changed, or how it should be changed to provide the added capability. An example is provided following the description.

To conduct an analysis of information needs and priorities, the information needs from different organization units/levels are summarized for each business function and process. To perform the analysis, the information need can be identified and defined as follows:

- The information need is named using a short name or sentence
- The information usage is stated (e.g., performing a business function, controlling an activity, etc.)
- The need is categorized—a distinction is made as to the level of decision-making the information supports, such as:
  - ◊ Strategic
  - ◊ Planning and analysis

- ◊ Monitoring and control
- ◊ Transaction

- The business object the need supports is identified:

  - ◊ Objective
  - ◊ Goal
  - ◊ Critical success factor
  - ◊ Performance measurement
  - ◊ Inhibitor

- The relative importance of the information need for the business function is determined:

  - ◊ Mission-critical
  - ◊ Very important
  - ◊ Important
  - ◊ Useful
  - ◊ Noncritical

- Support status showing whether the information is currently available or is planned is identified

- The type of information source is identified:

  - ◊ Computerized or manual
  - ◊ Internal or external

- The name of the system providing the information is identified

- The extent to which the current information satisfies the requirements is categorized:

  - ◊ Good
  - ◊ Moderate
  - ◊ Poor
  - ◊ No support

The following table provides an example of how this information could be presented in this section of the document.

Table 1. Example Requirement Documentation Technique

Need	Monitor customer queries
Usage	Performing a business function
Category (strategic, planning and analysis, monitoring and control, transaction)	Monitoring and control

Business object supported (objective, goal, critical success factor, performance measurement, inhibitor)	Performance measurement
Priority (mission critical, very important, important, useful, noncritical)	Very important
Availability (current or planned):	Current
Automated source?	Manual
Internal or external?	External
Source	Help Desk Information System
Current information satisfactory (good, moderate, poor, no support)?	Moderate

2.2 PERFORMANCE REQUIREMENTS

Time-dependent and nonfunctional requirements are described in this section. The same set of characteristics as those listed in the previous section are used with modifications where appropriate.

These requirements address performance as well as schedule, resource, hardware and software constraints. The following sections are repeated for each requirement using the same characteristics as above with modifications as required.

2.2.1 Constraints

Constraints address those environmental items that govern how the information system is developed. The specific restrictions must be sufficiently defined such that the impact on the project is evident.

2.2.2 Resource

Resources may include funding or personnel. Personnel resource constraints may include the availability of specific talents, the need for individual skill or category of personnel.

2.2.3 Hardware

In this section is described requirements for specific hardware, communications media and the need to interface with individual legacy or future systems.

2.2.4 Software

This section includes system software, programming environments, database or other supporting application packages. They may be required to implement the requirements or may represent legacy systems that the system must interface with when it is installed.

2.3 STANDARDS

If the organization requires that standards be followed, they are identified in this section. The standards can address documentation, implementation technologies, interfaces or any other topic addressed by the requirements.

2.4 LIMITATIONS

Any other limitations on the implementation of the requirements are described in this section.

2.5 OTHER REQUIREMENTS

Any other requirements not specifically covered in the functional, performance or non-functional requirements section are described here. The format to use in specifying these requirements must, of necessity, depend on the nature of the requirement. They should, however, adhere to the guidelines provided above to be sure that, insofar as possible, they are correct, unambiguous, verifiable and so on.

3 IMPACT OF REQUIREMENTS

This section documents the changes that are needed to meet the new requirements. Where appropriate this section must include revised or planned data store descriptions and revised or planned business process descriptions. Data flow diagrams are used to show the data and information flow between business systems, data stores and organizational work units. Both internal and external relationships should be shown. Examples of how to document this information can be drawn from the baseline information architecture.

4 ISSUES, RISKS AND ALTERNATIVES

This section documents the issues, risks and alternatives related to the implementation of the new requirements. The information can be organized by requirement or by issue, risk or alternative. Issues and risks should include information on how they can be resolved which may include the need for such analyses as a more formal risk analysis or feasibility study. Alternative implementations should be specified sufficiently to provide the Target Architecture team with a solid understanding of the parameters of the alternatives and what the critical decision making criteria are. There are a number of approaches for conducting analyses of alternatives that can be explored for use in this section.

C.12 TARGET ARCHITECTURE

Target Architecture Template

1 INTRODUCTION

When the baseline architecture needs to be modified and the new requirements have been documented, analysts must revise the existing architecture and design a new, or target architecture. The target architecture is described in this document.

1.1 PURPOSE

The purpose of the target architecture description document is to describe the desired new architecture for the enterprise which includes identification of all information systems, functions/entities and graphical descriptions of the architecture. This section of the document provides a summary of the reason for developing a target architecture. It may include a summary of the business need being met by the architecture and other issues that help to define the purpose.

1.2 SCOPE

This section describes the boundaries for the target architecture. It may be expressed in terms of timeframe, relationships to other systems or organizations, or functionality. This is a definition of the boundary between the individual systems and the environment of each functional area represented by the architecture. It is described at a high level with sufficient detail to clearly present the target architecture and explain the differences between it and the baseline architecture. The intended audience for this document is senior information technology managers.

1.3 BACKGROUND

The background section describes, at a high level, the current environment that will support and use the target architecture. It includes rationale and justification for the target architecture and as such may include goals, objectives and issues that contribute to the need for the target architecture, as well as a brief description of the business requirements that are the impetus for development of the target architecture.

1.4 REFERENCES

This section contains a list of documents related to the topics covered in this material. This may include detail level documents for individual information systems or business operations. The business requirements document that preceded development of the target architecture should be among those listed here.

1.5 ORGANIZATION OF THE DOCUMENT

Following the introduction and background, Section 2 describes the business requirements. The next three sections are the three target EISA views. Section 3 is the target work architecture, Section 4 is the target functional architecture and Section 5 is the target information architecture. The target technology infrastructure and information system descriptions are in Section 6.

The documents referenced in the following sections first appear as part of the baseline architecture document. These same documents provide the format for the components of the

target architecture and, thus, are referenced in this document as well. The difference between the use of the templates for the baseline and target architectures is that the target architecture reflects differences in the architectures motivated by new business requirements, mission statements or updates to the underlying technology which are documented in the business requirements document.

2 BUSINESS REQUIREMENTS

Documented in Appendix C.5.

3 TARGET WORK ARCHITECTURE

Documented in Appendix C.6.

4 TARGET FUNCTIONAL ARCHITECTURE

Documented in Appendix C.7.

5 TARGET INFORMATION ARCHITECTURE

Documented in Appendix C.8.

6 TARGET INFRASTRUCTURE

Documented in Appendix C.9.

Glossary
Index

C.13 ARCHITECTURAL DIFFERENCE DOCUMENT

Baseline/Target Architecture Differences Template

Title Page No.

1 INTRODUCTION

When a need to change the baseline architecture is identified and the new or target architecture is developed, the changes need to be documented. This document is the repository for information on which components of the baseline changed into which segments of the target architecture.

1.1 PURPOSE

The purpose of the architectural difference document is to identify and specify the differences between the baseline and target architectures. This information is critical to related tasks that update documentation, retrain staff and change administrative and/or management procedures to reflect the environment created by the target architecture.

1.2 SCOPE

This section describes the limits of the changes made to the baseline architecture with respect to implementation in the target architecture. The information presented here must be consistent with the scope section of the target architecture document. It should provide a clear understanding of the breadth and depth of the changes to the baseline architecture at a high level. The intended audience for this document are senior information technology managers.

1.3 BACKGROUND

This section summarizes the environmental, technical and functional framework within which the baseline architecture was assessed and the target architecture defined.

1.4 REFERENCES

This section contains a list of documents related to the topics covered in this material. This may include detail level documents for individual information systems or business operations. It should, at a minimum, reference the appropriate baseline and target architecture documents.

1.5 ORGANIZATION OF THE DOCUMENT

This section describes the organization of the document. A glossary and index should be included at the end of the document.

2 TECHNOLOGY ASSESSMENT

This section documents the differences between the baseline infrastructure and the target infrastructure. The descriptions should be drawn from the two documents and presented so that the difference between them is clear. It is suggested that this section repeat the graphics, text, tables and descriptive material whenever it is appropriate. The additional material added here is to highlight the essence of the change and to pinpoint the impacts of the change on the operations of the organization.

3 DESIGN CONSTRAINTS

This section documents the limitations on the target architecture design that are the result of such technical parameters as the availability or performance of technology or feasibility of a design approach. This information must be consistent with similar information presented in the target architecture document. These constraints may be described with respect to hardware, software or performance.

4 TRANSITION/MIGRATION OPPORTUNITIES

This section documents the analysis made to determine which of the target architecture components are feasible to implement in the next generation of the EISA. This provides an explanation of why the target architecture may be implemented in phases and/or why specific components are included earlier rather than later, or not at all.

4.1 CRITICAL DIFFERENCES

This section documents the differences between the baseline and target architectures identified during the gap analysis which have been classified as critical.

4.2 DECISION TAXONOMY

This section documents the analysis that was done to map the business, functional and operational requirements that conflict with each other or which have a positive or negative relationship with each other. These may be drawn from the business requirements document exclusively, or may be drawn from the existing baseline architecture and the business requirements document. The following example illustrates how this information can be presented.

Table 1. Example Requirements/Constraints Matrix

Constraints	Requirements			
		Interface QIS/ QA system	Enable min 2400 baud connectivity	Provide 8 hour repair capability
Hardware	1200 baud modem in 90 percent of facilities		X	
Software	QIS hosted on network database; QA system on relational database	X		
Performance	Current maintenance contract specifies 48 hour mean time to repair			X

4.3 FEASIBILITY ANALYSIS

This section of the document describes the alternatives for implementation of the target architecture, the analysis of those alternatives and the alternative that represents the best given the constraints.

4.3.1 Design Alternatives

This section describes, at a high level, the alternative designs for the target architecture implementation.

4.3.2 Alternative Number 1 (repeated as needed)

This section describes the alternative to implementation of the target architecture.

4.3.3 Cost Factors

This section describes the cost factors that impact the feasibility of the various alternatives. They may include all of the following and additional items as appropriate:

- Conversion
- Personnel
- Supplies
- Energy
- Maintenance
- Space

4.3.4 Non-Cost Factors

This section describes the non-cost factors that impact the feasibility of the various alternatives. They may include all of the following and any additional items as are appropriate:

- Functional
- Risk

4.3.5 The Most Feasible Alternative

The most feasible alternative is described in this section. It can be described using one of the following cost benefit analysis methods:

- Present value analysis
- Net present value
- Benefit-cost ratio

4.4 RISK ANALYSIS

This section documents the results of the analysis conducted to identify the risks associated with the target architecture project.

4.5 "QUICK HIT" OPPORTUNITIES

This section documents the projects that have been identified as deliverable in the short-term and providing high payoff. These provide the implementation team with opportunities to respond quickly to user needs and should be considered a high priority for implementation.

4.6 NEW INFORMATION SYSTEMS

This section describes the new information systems that will be provided by the target architecture. It should reference the components supporting the new systems through the target information, work, functional and infrastructure architectures. The description should be consistent with the information in the documentation for the target architectures.

Glossary
Index

C.14 ARCHITECTURE TRANSITION PLAN

Architecture Transition Plan (ATP) Template

1 INTRODUCTION

When the target architecture is designed and implementation is begun, the organization needs to develop a plan to transition to the new environment. This includes the process of implementing the new architecture and the transition from the use of baseline components to the use of target architecture components. This document establishes the boundaries of the transition project and the commitment of various organizations responsible for the planning and execution of the project.

1.1 PURPOSE

The purpose of the architecture transition plan is to describe the milestones and activities required to transition to the target architecture. It describes the steps needed to make the transition, roles and responsibilities, any impact on existing systems and the priority of the milestones and activities where applicable.

1.2 BACKGROUND

This section summarizes the background and provides the groundwork for the transition plan. It may include references to existing interim projects, published schedules for the development of the target architecture, or relevant milestones associated with existing baseline systems. The objective is to provide the context for the milestones and impact analysis described in later sections of this document.

1.3 SCOPE

This section defines boundaries of the transition and any limitations that may affect the successful completion of the project. It should be consistent with the scope of the baseline and target architectures.

2 ROLES AND RESPONSIBILITIES

This section describes the organization structure of the project. In particular, the assignments for the individual ADP team members (see the roster of ADP team members) must be defined. As described in the project plan, the individuals should be identified and associated with specific duties for the transition to the target architecture. The following table illustrates how this could be documented.

Project leader	Responsible for the direction and composition of transition plan activities
Requirements task lead	Responsible for being sure the transition plan addresses all the requirements affected by the target architecture
Information systems task lead	Responsible for being sure the transition plan addresses all information systems affected by the target architecture.
Infrastructure task lead	Responsible for being sure the transition plan addresses all the technical components required by the target architecture.

3 DEVELOPMENT MILESTONES

This section describes the milestones which will lead to a transition from the baseline to the target architecture. Milestones are important accomplishments that help determine progress in the transition from the baseline to the target architecture. The following example describes milestones that may be included in this section.

Table 1. Example Transition Milestone Chart

Milestone	Anticipated Due Date	Rationale
Relational query database	3 months after project start	Bridge between new GUI and legacy database until new OODB for queries is completed
New GUI for inventory database	4 months after project start	Provide updated access to legacy inventory system
Conversion of query database	5 months after project start	Migrate legacy database to OODB with new GUI

Any additional activities important to the transition should be included in this section.

4 IMPACT ON EXISTING SYSTEMS

It is important to understand the impact of the new architecture on existing systems. This is necessary to ensure that management can manage the change effectively and provide for the additional training, documentation and other assistance that may be needed for staff to move into the new environment efficiently and effectively. This step documents the impact of the target system architecture on the baseline architecture. The following table shows how existing information systems can be classified to show the impact.

Table 2. Example of Impact Analysis on Baseline Information System

Baseline Information System	Impact
Query information system	Completely replaced by OODB
Inventory system	Enhanced with new GUI

Glossary
Index

C.15 ARCHITECTURE IMPLEMENTATION PLAN

Architecture Implementation Plan (AIP) Template

1 INTRODUCTION

When the target architecture has been defined, a plan needs to be developed to manage the implementation of the new architecture. This document establishes the boundaries of the project plan and describes the roles and responsibilities of those responsible for the planning and execution of the implementation plan.

1.1 PURPOSE

The purpose of the implementation plan is to document the activities, required resources, the time required and the estimated cost to implement the target architecture. This document establishes the baseline from which progress will be measured in terms of cost and schedule. It is important that care be taken to plan the implementation in a realistic way so that the schedule is achievable within the budget allowed.

1.2 SCOPE

This section defines boundaries of the project and any limitations that may affect the successful completion of the project. Any budgetary limitations as well as an overview of the budget for the implementation project is also described in this section.

1.3 REFERENCES

Any background documents, policies, standards or procedures to be followed on the project are listed in this section. The transition plan should be included as well as the target architecture and any components of the baseline architecture that are relevant.

2 RISKS

Potential risks from a management, administrative and technical view should be described (such as a tight schedule, unavailability of personnel, limited funding, etc.). If the project is sufficiently large, this section should describe the overall approach to risk management and point to a separate risk management plan.

3 PERSONNEL

This section describes the organization structure of the project. In particular, the assignments for the individual implementation project team members. An organization chart for the project should be developed. Tables similar to the ones in the ADPP and the transition plan are appropriate to identify responsible staff and describe their roles in the implementation of the target architecture. The following example illustrates how these can be documented.

Staff Member	Role	Responsibilities
A. Jones	Project leader	Manage, direct and supervise tasks leading to the development and implementation of the target architectures
S. Wetherby	Baseline system liaison	Coordinate the activities of the implementation team with the staff who will be transitioning from the baseline architecture to the target architecture.

4 TASKS

This section describes the tasks for the implementation project. Each task is described in more detail in subsequent sections of the document. Each task description should follow the work breakdown structure included in the ADPP. Information to include is required input, expected output and an estimate of the level of effort in terms of staff hours. Project reviews and internal team reviews should be included. Of particular interest in this section are those tasks to develop required software components.

4.1 PROJECT TASKS

These are the administrative, management and supporting tasks required to implement the target architecture. They can be described in these categories depending on the complexity of the project. There should be a clear relationship to the roles and responsibilities described in an earlier section.

4.2 TASK A (REPEAT AS REQUIRED)

Task A: Develop a schedule for quality assurance reviews. This task includes development of agendas, notification of attendees, preparation of an evaluation checklist and solicitation of follow-up comments. The task should also develop criteria for promoting products to the next level of development and ways to handle products that are not deemed ready to move to the next level of development. The task will also coordinate with the configuration management task lead.

Input: Deliverables

Output: Quality assurance evaluation

Estimated level of effort: 3 staff hours every two weeks. This may be adjusted as required.

4.3 INFORMATION SYSTEM DEVELOPMENT TASKS

These tasks address those components of the target architecture that must be modified from the baseline, acquired through COTS, or created.

4.4 TASK AA (REPEAT AS REQUIRED)

Task: Convert Query Database: This task converts the existing query database to the object-oriented database format and reloads the data into the OODB.

Input: legacy data

Output: OODB

Estimated level of effort: 6 staff months

5 SCHEDULE

This section presents a comprehensive schedule for the transition from the baseline to the target architecture. All activities should be listed with a projected duration of time for their execution and milestone date when they should be completed. This allows management to determine whether the project is on time and to take steps to address schedule problems early in the project's life. It is highly recommended that an automated project planning software package be used for implementation projects of more than three to four months duration.

6 BUDGET

The budget for the implementation is documented in this section. Once the implementation is underway, the budget is updated with actuals. In this way, progress can be determined and management alerted if the project begins to overrun or underrun significantly. If a software cost estimating package is used, the output is documented in this section.

7 TECHNOLOGY AND RESOURCES

This section documents the software, hardware, communications and related technology which are required for the transition from the baseline to the target architecture. This includes facilities and data sources and any other resource necessary to the transition. The technology and resource requirements may be presented by task as described in section 4, or presented all together in this section.

This information is needed to be sure that all the enabling technology is acquired and accounted for as the implementation proceeds. If this information is tracked independently, management can be alerted to potential problems related to the acquisition or development of these products before they have too severe an impact on the project
In addition, a development library should be developed to store artifacts from the development effort and provide reference material for future development efforts. It can also serve as a repository for active and on-line information sources for the target architecture development team.

8 DOCUMENTATION AND PUBLISHING STANDARDS

This section describes the standards to be followed in preparing the documentation for the project. It also describes the standards to be followed in publishing deliverables and other output from the project. The standards are listed according to whether they are mandatory or optional.

9 CRITICAL SUCCESS FACTORS

When the target system has been implemented, the organization needs to be able to determine whether it is successful in terms of the goals and/or objectives originally offered to justify the project. This section of the plan should describe those factors that are to be met before the target architecture is considered a success. Insofar as possible, these factors should be measurable and an approach included on how to collect data to determine whether they have been met. The following table illustrated how this can be documented.

Critical Success Factor:	Improve the level of technology available to the Information Center staff
Objective:	Provide immediate access to high-speed printers to 75% of the staff
Measure	Number of staff with on-line access to high-speed printers following implementation of the target architecture

This is a measurable criteria and can be evaluated following implementation of the target architecture. A time frame should also be provided for achievement of these factors.

Glossary
Index

APPENDIX D
PROPOSING CHANGES TO THIS DOCUMENT

This appendix provides guidance for submission of proposed changes to the Treasury Architecture Development Process (TADP).

Each proposal should be described as specific wording for line-in/line-out changes to a specific part of the TADP. Use of a standard format for submitting a change proposal will expedite the processing of changes.

The preferred method of proposal receipt is via electronic mail. It is requested that change proposals be sent in ASCII format and be sent via the internet to the address below. If e-mail is not available, change proposals will be accepted in hard copy via surface mail or fax at the following address:

Ms. Jane Owens
Department of the Treasury
ODASIS/OIRM
Rm 2110
1425 New York Avenue, NW
Washington, DC 20005

Fax: 202-622-9147; please indicate TADP on the cover sheet

Internet: Asghar.Noor@cio.treas.gov

Changes submitted to ODASIS will be discussed in the Treasury Architecture Working Group of the CIO Council. Approved changes will be made to the document by ODASIS.

The format for submitting change proposal is as follows:

a. Point of Contact Identification

1. Name:

2. Organization and Office Symbol:

3. Street:

4. City:

5. State:

6. Zip Code:

7. Area Code and Telephone Number

8. Area Code and Fax Number

9. Electronic Mail Address:

b. Proposed Changes

1. Section Number:

2. Page Number:

3. Title of Proposed Change:

4. Wording of Proposed Change:

5. Rationale for Proposed Change:

6. Other Comments:

Please repeat items (1) through (6) for each proposed change.

The Section Number and Page Number fields allow the submitter to clearly identify where the proposed change should be made. The Title of the Proposed Change field is for the submitter to insert a brief title that gives a general indication of the nature of the proposed change. In the Wording of the Proposed Change field, the submitter should identify the specific words (or sentences) to be deleted and the exact words (or sentences) to be inserted. In this field, providing identification of the referenced paragraph, as well as the affected sentence(s) in that paragraph, would be helpful. The goal is for the commentator to provide proposed wording that is appropriate for insertion into the TADP without editing (i.e., a line-in/line-out change). The Rationale for the Proposed Change allows the commentator to provide additional commentary and support for making the proposed change. The more specific the rationale, the more likely the proposed change will be accepted. A statement concerning the impact of the proposed change may also be included in the rationale. Finally, any other information the commentator wishes to provide may be included in the Other Comments field. However, without some degree of specificity, these comments may not result in change to the document.